able, structured matching will not be computationally tractable, so some other task-specific technique will be needed. In practice, however, a large number of problems are statically decomposable or nearly so, as attested by the use of structured matching in many knowledge-based systems indicated previously.

V. Concluding Discussion

The search for task-specific knowledge-based problem solving techniques follows from the need to understand how intelligent problem solving is possible. Without sufficient constraints on how knowledge is organized and used, problem solving can easily become an intractable process. To these ends, we have formally described the task-specific technique of structured matching and discussed the conditions under which it is computationally feasible.

Our description of structured matching above is in terms of rules, tables, and hierarchies. Thus, structured matching at first glance might appear to be a straightforward combination of three familiar ideas in AI: production rules [13], decision tables [3], and hierarchical decomposition [14], [17]. However, since productions rules and decision tables are general enough to be Turing-universal, they do not ensure computational tractability. Moreover, hierarchical decomposition, without additional constraints, does not guarantee tractability [5]. Structured matching addresses the problem of computational feasibility by restricting the kinds of rules, tables, and hierarchies that are allowed in a structured matcher. For example, the rules in a structured matcher are only permitted to match specific inputs from "below." Tables are restricted to one kind of action—selecting a choice based on a small number of parameters. The hierarchy must partition the parameters into computationally manageable chunks. These constraints capture the essence of what makes a range of decision-making problems tractable to solve.

From the perspective of pattern recognition approaches, structured matching can be characterized as a heuristic technique. From this viewpoint, structured matching might appear to be an ad hoc approach because we do not provide an algorithm for constructing (optimal) structured matchers from a set of cases, but leave that construction as a problem of knowledge engineering. This reflects a difference of goals and perspectives. Our main emphasis in this paper is identifying structured matching as a useful construct for organizing domain and control knowledge for making decisions. Hence, our concern with the issues of computational efficiency and explicitness of representation.

The computational advantages of structured matching are due to a coupling between an information-processing task and a knowledge-based technique. The technique of structured matching is specific to the select-1-out-of-n task, and it allows decision-making knowledge to be clearly represented and efficiently applied. We believe that this coupling between techniques and tasks will explain much about how intelligent decision making is possible.

Acknowledgment

This work has benefited from many discussions with B. Chandrasekaran, T. Johnson, J. Josephson, J. Smith, and N. Soundarajan. Of course, we are responsible for any remaining errors.

References


A Simplex-Like Algorithm for the Relaxation Labeling Process

XINHUA ZHUANG, ROBERT M. HARALICK, AND HYONAM JOO

Abstract—In this correspondence a simplex-like algorithm is developed for the relaxation labeling process. The algorithm is simple and has a fast convergence property which is summarized as one more step theorem. The algorithm is based on fully exploiting the linearity of the variational inequality and the linear convexity of consistent labeling search space, somewhat similar to the simplex algorithm in linear programming.

Index Terms—Consistent labeling, linear programming, relaxation, simplex algorithm, variational inequality.

I. Introduction

R. A. Hummel and S. W. Zucker [1] developed a theory to explain what relaxation labeling accomplishes. The theory is based on fully exploiting the linearity of the variational inequality and the linear convexity of consistent labeling search space, somewhat similar to the simplex algorithm in linear programming.

Manuscript received May 14, 1986; revised July 31, 1989. Recommended for acceptance by S. W. Zucker.

The authors are with the Department of Electrical Engineering, University of Washington, Seattle, WA 98195.

IEEE Log Number 8931122.
on an explicit new definition of consistency in terms of variational inequality, and leads to a relaxation algorithm with an updating formula which uses a projection operator.

In this correspondence, based on fully exploiting the linearity of the variational inequality and the linear convexity of consistent labeling search space, an essential characterization of a consistent labeling is obtained (see Section III), the structure of the consistent labeling set becomes clear (see Section IV) and hence an efficient simplex like algorithm is developed (see Section V). The convergence of the algorithm is explored in Section VI (see Theorem 6.1-2). Theorem 6.1 carries the name "one more step theorem" which indicates the algorithm takes the shortest path. A comparison to the Rosenfeld et al. consistent labeling definition is made in Section VII. In addition, an essential condition of a consistent labeling in Rosenfeld et al. is derived. It is also proved that a consistent labeling in Rosenfeld et al. is implied by Hummel and Zucker, and hence there exists a consistent labeling in Rosenfeld et al. The experimental results which are given in the final section, Section VIII, verify the theory and algorithm developed in the correspondence. For readers' convenience, a specific section (Section II) for notation and definition is included as well.

II. NOTATION AND DEFINITIONS

A consistent labeling problem has units each of which has an unknown true label. There are n units, denoted by \( U_1, \ldots, U_n \), and \( m \) labels, denoted by \( L_1, \ldots, L_m \). Each \( U_i \) (\( i = 1, \ldots, n \)) will be assigned a set of \( m \) numbers \( p_i(1), \ldots, p_i(m) \) called a labeling distribution:

\[
p_i(1) \geq 0, \ldots, p_i(n) \geq 0,
\]

\[
\sum_{j=1}^{m} p_j(i) = 1.
\]

For abbreviation, we let

\[
p(i) = (p(i)_1, \ldots, p(i)_m),
\]

\[
i = 1, 2, \ldots, n,
\]

and simply call the \((1 \times m)\)-row vector \( p(i) \) a labeling distribution of \( U_i \).

Between label assignments there are consistency constraints. Let a real number \( r(i,j; h,k) \) represent how the label \( L_h \) at the unit \( U_i \) influences the label \( L_k \) at the unit \( U_j \). If the unit \( U_i \) having the label \( L_h \) lends a high support to the unit \( U_j \) having the label \( L_k \), then \( r(i,j; h,k) \) should be large and positive. If constraints are such that the unit \( U_i \) having the label \( L_h \) means that the label \( L_k \) at the unit \( U_j \) is highly unlikely, then \( r(i,j; h,k) \) should be small. No specific restrictions are placed on the magnitude of \( r(i,j; h,k) \). However, we do require that

\[
r(i; j; h) = r(i, j, h) = r(j, i, h) = r(i, j, k) = r(i, j, k; h) = r(i, j, k; h),
\]

\[
\alpha \leq r(i, j; h, k) \leq \alpha,
\]

\[
r(i, j; h, k) = \alpha \text{ constant, for instance } \alpha,
\]

\[
\text{independent of } i, j,
\]

\[
r(i, j; h, k) \leq \alpha.
\]

Define the support on the unit \( U_i \) having the label \( L_j \), \( \{ U_i, L_j \} \), from the unit \( U_i \) having the label \( L_h \) with a labeling distribution component \( p_i(h) \), \( \{ U_i, L_h, p_i(h) \} \), by \( r(i, j; h, k) \).

Define the support on the unit \( U_i \) having the label \( L_j \) with another labeling distribution component \( p_j(h) \), \( \{ U_i, L_j, p_j(h) \} \), from \( U_i \) having labels \( L_h \) and \( L_k \) by \( r(i, j; h, k) \).

Define the support on \( \{ U_i, L_j, p_j(h) \} \) from \( U_i \) having a label distribution \( p(h) \), \( \{ U_i, p(h) \} \), by \( \sum_{j=1}^{m} r(i, j; h, k) p_j(h) \).

Define the support on \( \{ U_i, L_j, v(i) \} \) from \( n \) labeling distributions, \( p(1), \ldots, p(n) \), by \( \sum_{j=1}^{m} r(i, j; h, k) p_j(h) \).

Define the support on \( \{ U_i, v(i) \} \) from \( P \) by \( \sum_{j=1}^{m} q(i, P) v_j(i) \).

Define the support on \( n \) labeling distributions, \( v(1), \ldots, v(n) \), or \( V = \{ v(1), \ldots, v(n) \} \) from \( P \) by \( \sum_{i=1}^{n} q(i; P) v_i(i) \).

For abbreviation, we let

\[
q(i; P) \triangleq \{ q(i; P), \ldots, q_n(i; P) \},
\]

\[
q(P) \triangleq \{ q(1; P), \ldots, q(n; P) \}
\]

and simply call each of \( P \) and \( V \) a labeling. Thus the support on \( V \) from the labeling \( P \) is represented by the inner product \( (q(P), V) \) in the \( n \)-dimensional Euclidean space \( E_n \):

\[
(q(P), V) = \sum_{i=1}^{n} (q(i; P), v(i)),
\]

where each \( (q(i; P), v(i)) \) represents the inner product in the \( m \)-dimensional Euclidean space \( E_m \).

A set of \( n \) labeling distributions \( p(1), \ldots, p(n) \) is called unambiguous if each of \( n \) units is assigned a unique label, that is, for each \( i, 1 \leq i \leq n \), all \( p_i(j) \)'s \( (j = 1, \ldots, m) \) are zero except one which is \( 1 \). R. A. Hummel and S. W. Zucker first define a consistency concept of an unambiguous labeling, then by analogy they define a consistency concept of an ambiguous labeling. According to their definition, \( n \) labeling distributions \( p(1), \ldots, p(n) \) comprise a consistent labeling if for various \( n \) labeling distributions \( v(1), \ldots, v(n) \) there hold the following variational inequalities:

\[
(q(i; P), v(i) - p(i)) \leq 0, \quad i = 1, \ldots, n,
\]

or the same

\[
(q(i; P), v(i)) \leq (q(i; P), p(i)), \quad i = 1, \ldots, n.
\]

In other words, \( P \) is a consistent labeling if and only if \( v(i) \) maximizes \( q(i; P), v(i) \) when \( v(i) \) varies over \( K \) [see (14)]. It is clear that a consistent labeling \( P \) gives the support in favor of itself or discriminates against any other labelings since

\[
(q(P), V) = \sum_{i=1}^{n} (q(i; P), v(i)) \leq \sum_{i=1}^{n} (q(i; P), p(i)) = (q(P), p).
\]

Conversely, if a labeling \( P \) gives the support in favor of itself, i.e., for any other labeling \( V \) it holds that

\[
(q(P), V) \leq (q(P), P),
\]

then the labeling \( P \) is consistent, i.e., for each \( i, 1 \leq i \leq n, (10) \) holds, since letting each \( v(h) \) equal \( p(h) \) except \( v(i) \) which could be arbitrary, (12) will imply (10), as easily verified. Thus, a consistent labeling \( P \) could also be defined by the single variational inequality, i.e., (12). In other words, \( P \) is a consistent labeling if and only if \( P \) maximizes \( (q(P), V) \) when \( V \) varies over \( K \) [see (16)].

Let \( e_1, \ldots, e_m \) be \( m \) standard basic vectors in \( E_m \). Let

\[
K_0 = \{ e_1, \ldots, e_m \}.
\]

\[
K = \{ \sum_{j=1}^{m} u_j e_j : u_j \geq 0, \sum_{j=1}^{m} u_j = 1 \}.
\]

Then \( K(K^n) \) is a linear convex set in \( E_n \) and \( K_0(K^n) \) the set of vertices of \( K(K^n) \). The set \( K \) takes a specific name "simpex" in topology and linear programming. It is clear that \( q(P), v(i) \) defines a linear transformation: \( K^n \rightarrow E_n \) and \( q(i; P) \) a linear functional: \( K^n \rightarrow E \). The inner product \( (q(P), V) \) defines a bilinear functional: \( K^n \times K^n \rightarrow E \) and the inner product \( (q(i; P), v(i)) \) a bilinear functional: \( K^n \times K^n \rightarrow E_1 \).

Hummel and Zucker call a labeling \( P \) strictly consistent if for
each $v(i) \in K$, $v(i) \neq p(i)$, it holds that

$$q(i; P), v(i) < q(i; P), p(i)) \quad i = 1, \ldots, n. \quad (17)$$

In other words, $P$ is a strictly consistent labeling if and only if for all $v, i = 1, \ldots, n, p(i)$ is a unique maximal point of $(q(i; P), v(i))$ when $v(i)$ varies over $K$. Similarly, it could be proved that a labeling $P$ is strictly consistent if and only if for each $V \in K^n$, $V \neq P$, it holds that

$$q(P), V < q(P), P. \quad (18)$$

In other words, $P$ is strictly consistent if and only if $P$ is a unique maximal point of $(q(P), V)$ when $V$ varies over $K^n$.

### III. Characterization of a Consistent Labeling

To characterize a consistent labeling it is natural and reasonable to exploit the linearity of variational inequalities (9) and the convexity of the labeling distribution search space $K$.

The consistency condition suggests that to find a consistent labeling $P = \{p(1), \ldots, p(n)\}$ with $p(i) = (p_i(i), \ldots, p_m(i))$ we need first to consider

$$\max_{v \in K} (q(i; P), v(i)) \quad i = 1, \ldots, n. \quad (19)$$

Each maximum will be reached at vertices of $K$ since the inner product $(q(i; P), v(i))$ is linear w.r.t. $v(i)$ and the search space $K$ is a linear convex set. Let $M_0(i; P)$ be the set of the vertices which correspond to components of $q(i; P)$ where the component of $q(i; P)$ attains the maximum value:

$$M_0(i; P) = \{e_i; (q(i; P), e_i) = \max_{1 \leq k \leq m} (q(i; P), e_k)\}. \quad (20)$$

Let $M(i; P)$ be the linear convex set having $M_0(i; P)$ as its vertex set. Then it is clear that $M(i; P)$ is a face of $K$ and represents the maximal point set. That is

$$M(i; P) = \{u(i); (q(i; P), u(i)) = \max_{v \in K} (q(i; P), v(i))\}. \quad (21)$$

From definition (20), it is easy to derive that

$$M_0(i; P) = \{e_i; q_i(i; P) = \max_{1 \leq k \leq m} q_k(i; P)\} \quad (22)$$

and hence

$$M(i; P) = \left\{u; q_i(i; P) \geq \sum_{j=1}^m u_j, u_j \geq 0, \sum_{j=1}^m u_j = 1, u_i = 0 \right\} \quad (23)$$

Since $P$ is a consistent labeling if and only if for all $v, i = 1, \ldots, n, p(i)$ is a maximal point of $(q(i; P), v(i))$, when $v(i)$ varies over $K$, we can now characterize a consistent labeling $P$ by:

$$P(i) \in M(i; P), \quad i = 1, \ldots, n. \quad (24)$$

Since $P$ is a strictly consistent labeling if and only if for all $v, i = 1, \ldots, n, p(i)$ is a unique maximal point of $(q(i; P), v(i))$ when $v(i)$ varies over $K$, we can characterize a strictly consistent labeling $P$ by:

$$M(i; P) = M_0(i; P) = \{p(i)\} \quad i = 1, \ldots, n. \quad (25)$$

In this case each $p(i)$ must be a vertex of $K$ and hence a strictly consistent labeling is unambiguous.

Let

$$M_0(1; P) \times \cdots \times M_0(n; P) \quad (26)$$

and

$$M(1; P) \times \cdots \times M(n; P) \quad (27)$$

Then we can also characterize a consistent labeling $P$ by:

$$P \in M(1; P). \quad (28)$$

and a strictly consistent labeling $P$, which must be a vertex of $K^n$, by:

$$M(P) = M_0(P) = \{P\}. \quad (29)$$

It is understandable from a practical point of view that strictly consistent labelings are preferable because they are unambiguous and isolated. The latter will be explained in the next section.

### IV. Structure of the Consistent Labeling Set

From Kinderlehrer and Stampacchia [2] we know that the consistent labeling set denoted by $Z$ is nonempty. Obviously, $Z$ is a compact set in $K^n$. In this section we will explore the structure of $Z$. When is a consistent labeling isolated? Does the consistent labeling set have "linearity" and "convexity"? Or, when can two consistent labelings be connected by "a line segment" on which each point is a consistent labeling? Here $P \in Z$ is called isolated if $\forall V \notin P, \|V - P\| \leq \epsilon (\epsilon \approx 0)$.

We need the following important properties of $M(i; P)$:

For any fixed labeling $P^0$, it always holds that

$$M(i; P) \subset M(i; P^0), \quad i = 1, \ldots, n. \quad (30)$$

or briefly

$$M(P) \subset M(P^0), \quad (31)$$

whenever $\|P - P^0\|$ is small.

From any two labelings $P^0$ and $P^+$, it holds that

$$M(i; P^0) \cap M(i; P^+) \subset M(i; tP^0 + (1 - t)P^+), \quad i = 1, \ldots, n, \quad (32)$$

or briefly

$$M(P^0) \cap M(P^+) \subset M(tP^0 + (1 - t)P^+), \quad (32)'$$

where $0 \leq t \leq 1$. To prove (31)-(32) we need only to prove that

$$M_0(i; P) \subset M_0(i; P^0), \quad i = 1, \ldots, n, \quad (33)$$

and hence $V \notin Z$. Whenever $\|P - P^0\|$ is small, and

$$M_0(i; P^0) \cap M_0(i; P^+) \subset M_0(i; tP^0 + (1 - t)P^+), \quad t = 1, \ldots, n, \quad (34)$$

where $0 \leq t \leq 1$.

Suppose $e_i \notin M_0(i; P^0)$. We are to verify that $e_i$ does not belong to $M_0(i; P)$ either whenever $\|P - P^0\|$ is small. As a matter of fact, $e_i \notin M_0(i; P^0)$ means that for all $q_k(i; P^0)$ we have $q_k(i; P^0) < q_k(i; P^+)$. Because of continuity, however, $q_k(i; P) < q_k(i; P^0)$ immediately follows whenever $\|P - P^0\|$ is small. Thus, $e_i$ will not belong to $M_0(i; P)$. That validates (33).

Suppose $e_i \in M_0(i; P^0) \cap M_0(i; P^+)$. Then $P^0$ have

$$q_k(i; P^0) \geq q_k(i; P^+), \quad q_k(i; P^+) > q_k(i; P^0), \quad (35)$$

and hence $P^0$ is a linear functional w.r.t. $P^0$. The last inequality implies $e_i \in M_0(i; tP^0 + (1 - t)P^+)$. That validates (34).

By means of (31) or (31)' we can prove that a strictly consistent labeling is isolated. Suppose $P^0$ is a strictly consistent labeling. Then $M(P^0) = \{P^0\}$ by (29) and furthermore $M(P) = \{P^0\}$ whenever $\|P - P^0\| \leq \epsilon$ by (31). Thus $P^0 \neq P$ and $\|P - P^0\| < \epsilon$, we obtain $P \notin M(P)$, which implies $P \notin Z$. It verifies that $P^0$ is isolated.

By means of (32) or (32)' we can prove that if $P^0, P^+ \in Z$ and $M(P^0) \cap M(P^+) \neq \phi$, then $\forall t, 0 \leq t \leq 1, tP^0 + (1 - t)P^+ \in Z$, which means $Z$ is a connected set.
Z if and only if for some \( t_0 \), \( 0 < t_0 < 1 \), \( t_0 P' + (1 - t_0) P'' \in Z \) if and only if 
\[
(q(P' - P''), P' - P'') = 0.
\]
Suppose \( P \in M(P') \cap M(P'') \). Then \( P \in M(tP' + (1 - t)P'') \) \((0 < t < 1)\) by (32). This means that \( P \) is a maximal point of 
\[
(q(tP' + (1 - t)P''), V)
\]
as \( V \) varies over \( K'' \). Now requiring \( tP' + (1 - t)P'' \in Z \) is equivalent to requiring 
\[
tP' + (1 - t)P'' \in M(tP' + (1 - t)P'').
\]
In other words, \( tP' + (1 - t)P'' \) should be a maximal point of 
\[
(q(tP' + (1 - t)P''), V).
\]
Thus the necessary and sufficient condition for \( tP' + (1 - t)P'' \in Z \) will be:
\[
(q(tP' + (1 - t)P''), P') = (q(tP' + (1 - t)P''), P'') = 0,
\]
or after manipulations
\[
0 = t(q(P'), P - P') + (1 - t)^2 (q(P''), P - P')
\]
and for any \( v(i) \in M(i; P) \)
\[
(v(i), w(i; P) - p(i)) = \sum_{j, r \in M(i; P)} v_j(i) \left[ w_j(i; P) - p_j(i) \right]
\]
which is independent of \( v(i) \). Therefore, \( w(i; P) \) belongs to \( M(i; P) \) and \( M(P) \) comprises the unique orthogonal projection of \( p(i) \) onto \( M(i; P) \).

Now we are able to summarize the algorithm.

Step 1. Set \( P' \).
Step 2. Set \( k = 1 \).
Step 3. Compute \( M_k(P') \).
Step 4. Compute \( P^{k+1} = W(P^k) \).
Step 5. If \( (P^{k+1} = P^k) \) stop.
Step 6. Set \( k = k + 1 \).
Step 7. Go To Step 3.

The algorithm has a geometric explanation—something like

1) Start at \( P \).
2) Compute \( q(i; P), i = 1, \ldots , n \).
3) For each \( i \), change \( p(i) \) to lie on the face (or vertex) determined by \( q(i; P) \).
Repeat 2.3 until no change.

The next section is devoted to a convergence discussion.

VI. CONVERGENCE DISCUSSION

As seen, the proposed algorithm is simple and easily implementable. It has also nice convergence properties since the linearity of variational inequalities and linear convexity of the consistent labeling search space are exploited. The following Theorem 6.1 is something similar to the local convergence theorem by Hummel and Zucker, but it is a little bit nicer. It confirms that the algorithm finds the shortest path: when it starts with a point close to a strictly consistent labeling, only one more iteration is needed to reach the goal. Theorem 6.2 relates that any sequence produced by the algorithm, if it converges, must converge to a consistent labeling.

**Theorem 6.1. (One More Step Theorem):** Assume \( P'' \) is a strictly consistent labeling. Then, when \( P_k \) is close to \( PO \), only one more iteration is needed to reach the goal \( PO \). That is,
\[
P^{k+1} = PO.
\]

**Proof:** Since \( P'' \) is a strictly consistent labeling, \( M(P) \) will consist of a single point \( PO \), whenever \( \| P - PO \| \) is small, as argued before. Thus, when \( P^k \) is close to \( PO \), it holds that
\[
M(P^k) = \{ P^0 \},
\]
which implies that
\[
P^{k+1} = W(P^k) = PO.
\]
since the orthogonal projection of \( P^k \) onto \( \{ P^0 \} \) equals \( P^0 \).

**Q.E.D.**

**Theorem 6.2:** If the sequence \( \{ P^k \} \), produced by the algorithm, approaches \( P^0 \), then \( P^0 \) is a consistent labeling.

**Proof:** Since \( P^k \) approaches \( P^0 \), there is a \( k_0 \) such that
\[
M(P^k) \subset M(P^0), \quad k \geq k_0.
\]
which is implied by (31).

According to the algorithm, \( P^{k+1} \), being the orthogonal projection of \( P^k \) onto \( M(P^0) \), should belong to \( M(P^0) \) and hence \( M(P^0) \) as \( k \geq k_0 \). Since \( W(P^0) \) represents the orthogonal projection of \( P^0 \) onto \( M(P^0) \), for any \( P \in M(P^0) \), it holds that
\[
\| P - P^0 \| \geq \| W(P^0) - P^0 \|.
\]
which concludes especially when junctions of a triangle shown in [3, Fig. 1].

* 0.

Suppose labeling. Then for each \( p_j(i) \)

which means that for each \( p_j(i) > 0 \), \( q_j(i; P) \) keeps constant, independent of \( j \). We leave the easy proof with readers. Using the characterization, we could prove that Hummel and Zucker's consistent labeling set is nonempty as well.

\[ r(i, j; \lambda, \lambda') = d_{ij} : q_j(i, \lambda, \lambda') \]

where the \( d_{ij} \)'s are constant coefficients. Then, the function \( q_j^{(i)}(\lambda) \) which is the change in \( p_j^{(i)}(\lambda) \) in the \( k \)th iteration, where \( q_j^{(i)}(\lambda) \) are the notation used in [3], is same as the support function \( q_j(i; P) \) in the new algorithm. Using the same values for \( r_j(i, \lambda, \lambda') \) and \( d_{ij} \) as Rosenfeld et al. used in their example, two experiments have been performed as follows.

**A. Labeling Lines of a Triangle**

The problem is to label three units \( U_i(i = 1, 2, 3) \), three sides of a triangle, with four line labels \( l_i(i = 1, \ldots, 4) \), the set of four line labels \( \{ +, -, \pm, \rightarrow \} \) used by Waltz (see [4]). The behavior of the label distributions for the algorithm proposed by Rosenfeld et al. (Algorithm 1) and the one proposed in this paper (Algorithm 2) is illustrated in Fig. 1 for various initial labeling distributions. The row vector of each matrix in the figure represents the labeling distribution for each unit.

**B. Labeling Junctions of a Triangle**

In this example, the three junctions of a triangle are considered as units. There are six allowable L-junction types (labels). The six labels can be described by the line label pairs as: \{ \( (\rightarrow, \rightarrow), (\rightarrow, \rightarrow), (\rightarrow, \rightarrow), (\rightarrow, \rightarrow), (\rightarrow, \rightarrow) \} \}. The behavior of the label distributions for Algorithm 1 and Algorithm 2 is illustrated in Fig. 2 for various initial labeling distributions.

For the line labeling case, using Algorithm 2, the first iteration in Case A gives

\[
\begin{array}{cccccc}
0.5 & 0.5 & 0 & 0 \\
0.5 & 0.5 & 0 & 0 \\
0.5 & 0.5 & 0 & 0 \\
\end{array}
\]

The compatibility of label \( \lambda \) on unit \( a_i \) with label \( \lambda' \) on unit \( a_j \), \( r_j(i, \lambda, \lambda') \), is related to the function \( r(i, j; \lambda, \lambda') \) in this paper as:

\[
q_j(i, \lambda, \lambda') = d_{ij} \cdot q_j(i, \lambda, \lambda')
\]

The simple example of scene labeling considered by Rosenfeld et al. (see [3]) is used to verify the new relaxation algorithm developed in this paper. The problem is to label either the line or the junctions of a triangle shown in [3, Fig. 1].

The compatibility of label \( \lambda \) on unit \( a_i \) with label \( \lambda' \) on unit \( a_j \), \( r_j(i, \lambda, \lambda') \), is related to the function \( r(i, j; \lambda, \lambda') \) in this paper as:

\[
q_j(i, \lambda, \lambda') = d_{ij} \cdot q_j(i, \lambda, \lambda')
\]

The simple example of scene labeling considered by Rosenfeld et al. (see [3]) is used to verify the new relaxation algorithm developed in this paper. The problem is to label either the line or the junctions of a triangle shown in [3, Fig. 1].

<table>
<thead>
<tr>
<th>Case</th>
<th>Initial Distributions</th>
<th>Algorithm 1 after 75 iterations</th>
<th>Algorithm 2 after 1 iteration</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.25 0.25 0.25 0.25</td>
<td>0.27 0.27 0.23 0.23</td>
<td>0 0 0 0</td>
</tr>
<tr>
<td></td>
<td>0.25 0.25 0.25 0.25</td>
<td>0.27 0.27 0.23 0.23</td>
<td>0 0 0 0</td>
</tr>
<tr>
<td></td>
<td>0.25 0.25 0.25 0.25</td>
<td>0.27 0.27 0.23 0.23</td>
<td>0 0 0 0</td>
</tr>
<tr>
<td></td>
<td>0 0.5 0.5 0.5</td>
<td>0.99 0 0.01 0</td>
<td>1 0 0 0</td>
</tr>
<tr>
<td></td>
<td>0 0.5 0.5 0.5</td>
<td>0.99 0 0.01 0</td>
<td>1 0 0 0</td>
</tr>
<tr>
<td></td>
<td>0.5 0.5 0.5 0.5</td>
<td>0 0.99 0.01 0</td>
<td>1 0 0 0</td>
</tr>
<tr>
<td></td>
<td>0.5 0.5 0.5 0.5</td>
<td>0 0.99 0.01 0</td>
<td>1 0 0 0</td>
</tr>
<tr>
<td></td>
<td>0.5 0.5 0.5 0.5</td>
<td>0.99 0 0.01 0</td>
<td>1 0 0 0</td>
</tr>
<tr>
<td></td>
<td>0.5 0.5 0.5 0.5</td>
<td>0.99 0 0.01 0</td>
<td>1 0 0 0</td>
</tr>
<tr>
<td></td>
<td>0.5 0.5 0.5 0.5</td>
<td>0.99 0 0.01 0</td>
<td>1 0 0 0</td>
</tr>
<tr>
<td></td>
<td>0.5 0.5 0.5 0.5</td>
<td>0.99 0 0.01 0</td>
<td>1 0 0 0</td>
</tr>
<tr>
<td></td>
<td>0.5 0.5 0.5 0.5</td>
<td>0.99 0 0.01 0</td>
<td>1 0 0 0</td>
</tr>
<tr>
<td></td>
<td>0.5 0.5 0.5 0.5</td>
<td>0.99 0 0.01 0</td>
<td>1 0 0 0</td>
</tr>
<tr>
<td></td>
<td>0.5 0.5 0.5 0.5</td>
<td>0.99 0 0.01 0</td>
<td>1 0 0 0</td>
</tr>
<tr>
<td></td>
<td>0.5 0.5 0.5 0.5</td>
<td>0.99 0 0.01 0</td>
<td>1 0 0 0</td>
</tr>
<tr>
<td></td>
<td>0.5 0.5 0.5 0.5</td>
<td>0.99 0 0.01 0</td>
<td>1 0 0 0</td>
</tr>
<tr>
<td></td>
<td>0.5 0.5 0.5 0.5</td>
<td>0.99 0 0.01 0</td>
<td>1 0 0 0</td>
</tr>
<tr>
<td></td>
<td>0.5 0.5 0.5 0.5</td>
<td>0.99 0 0.01 0</td>
<td>1 0 0 0</td>
</tr>
<tr>
<td></td>
<td>0.5 0.5 0.5 0.5</td>
<td>0.99 0 0.01 0</td>
<td>1 0 0 0</td>
</tr>
<tr>
<td></td>
<td>0.5 0.5 0.5 0.5</td>
<td>0.99 0 0.01 0</td>
<td>1 0 0 0</td>
</tr>
<tr>
<td></td>
<td>0.5 0.5 0.5 0.5</td>
<td>0.99 0 0.01 0</td>
<td>1 0 0 0</td>
</tr>
<tr>
<td></td>
<td>0.5 0.5 0.5 0.5</td>
<td>0.99 0 0.01 0</td>
<td>1 0 0 0</td>
</tr>
<tr>
<td></td>
<td>0.5 0.5 0.5 0.5</td>
<td>0.99 0 0.01 0</td>
<td>1 0 0 0</td>
</tr>
<tr>
<td></td>
<td>0.5 0.5 0.5 0.5</td>
<td>0.99 0 0.01 0</td>
<td>1 0 0 0</td>
</tr>
<tr>
<td></td>
<td>0.5 0.5 0.5 0.5</td>
<td>0.99 0 0.01 0</td>
<td>1 0 0 0</td>
</tr>
<tr>
<td></td>
<td>0.5 0.5 0.5 0.5</td>
<td>0.99 0 0.01 0</td>
<td>1 0 0 0</td>
</tr>
</tbody>
</table>

**Fig. 1.** Experimental result of the line labeling.
and the second iteration in the same case gives

\[
\begin{array}{cccc}
0 & 0 & 0.5 & 0.5 \\
0 & 0 & 0.5 & 0.5 \\
0 & 0 & 0.5 & 0.5 \\
\end{array}
\]

Afterward, the results are oscillative. However, Algorithm 1 after 25 iterations gives

\[
\begin{array}{cccc}
0.27 & 0.27 & 0.23 & 0.23 \\
0.27 & 0.27 & 0.23 & 0.23 \\
0.27 & 0.27 & 0.23 & 0.23 \\
\end{array}
\]

It seems both algorithm do not give a meaningful interpretation in Case A. In cases B, C, E, and G both algorithms give the most probable interpretation. In case I both algorithms give the desired interpretation. In cases D and F two algorithms give different interpretations. However Algorithm 2 gives the most probable interpretation. In all cases except case A Algorithm 2 takes only one iteration to reach the goal in comparison to more than 25 iterations required by Algorithm 1.

For the junction labeling case, both algorithms give the same most probable interpretation in cases A and D. In cases B and C, Algorithm 1 gives an appropriate interpretation and Algorithm 2 gives the most probable interpretation. In case E Algorithm 1 gives another appropriate interpretation and Algorithm 2 gives the most probable interpretation. In case F Algorithm 1 gives an ambiguous result and Algorithm 2 gives the most probable interpretation. In cases G, H, and I both algorithms give the same appropriate interpretation. In case I both algorithms give the same another appropriate interpretation. In all cases, Algorithm 2 takes one or two iterations to reach the goal instead of taking more than 25 iterations required by Algorithm 1.

*Fig. 2. Experimental result of the junction labeling.*

The authors are thankful to the reviewers for their comments and suggestions.

**Acknowledgment**

The authors are thankful to the reviewers for their comments and suggestions.

**References**


