ble, 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 [6]. 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

[1] T. Bylander, A. Goel, and T. R. Johnson, "Structured matching: A computationally feasible technique for making decisions," Lab. for AI Res., Dep. CIS, Ohio State Univ., Columbus, Tech. Rep., June 1988.
[2] T. Bylander and S. Mittal, "CSRL: A language for classificatory problem solving and uncertainty handling," *AI Mag.*, vol. 7, no. 3, pp. 66-77, 1986.
[3] H. N. Cantrell, J. King, and F. E. H. King, "Logic structure tables," *Commun. ACM*, vol. 4, no. 5, pp. 272-275, 1961.
[4] B. Chandrasekaran, "Generic tasks in the knowledge-based reasoning: High-level building blocks for expert system design," *IEEE Expert*, vol. 1, no. 3, pp. 23-30, 1986.
[5] B. Chandrasekaran and S. Mittal, "Conceptual representation of medical knowledge for diagnosis by computer: MDX and related systems," in M. Yovits, Ed., *Advances in Computers, Vol. 22.* New York: Academic, 1983, pp. 217-293.
[6] D. Chapman, "Planning for conjunctive goals," *Artificial Intell.*, vol. 32, no. 3, 1987.
[7] E. Charniak, C. K. Riesbeck, D. V. McDermott, and J. R. Meehan, *Artificial Intelligence Programming*, 2nd ed. Hillsdale, NJ: Lawrence Erlbaum, 1987.
[8] W. J. Clancey, "Heuristic classification," *Artificial Intell.*, vol. 27, no. 3, pp. 289-350, 1985.
[9] A. Goel, B. Chandrasekaran, and D. A. Sylvan, "JESSE: An information processing model of policy decision making," in *Proc. 3rd Expert Syst. in Gov. Conf.*, Washington DC, Oct. 1987, pp. 178-187.
[10] A. Goel., N. Soundararajan, and B. Chandrasekaran, "Complexity in classificatory reasoning," in *Proc. 6th Nat. Conf. Artificial Intell.*, Seattle, WA, 1987, pp. 421-425.
[11] T. R. Johnson, J. W. Smith, and T. Bylander, "HYPER—Hypothesis matching using compiled knowledge," in *Proc. AAMSI Congr.*, San Francisco, CA, May 1989.
[12] J. McDermott, "Preliminary steps toward a taxonomy of problem solving methods," in S. Marcus, Ed., *Automating Knowledge Acquistion for Expert Systems.* Norwell, MA: Kluwer Academic, 1988.
[13] A. Newell and H. A. Simon, *Human Problem Solving.* Englewood Cliffs, NJ: Prentice-Hall, 1972.
[14] E. D. Sacerdoti, *A Structure for Plans and Behavior.* New York: Elsevier, 1977.
[15] A. L. Samuel, "Some studies in machine learning using the game of Checkers II—Recent progress," *IBM J.*, vol. 11, no. 11, pp. 601-617, 1967.
[16] S. Shum, J. Davis, W. F. Punch, and B. Chandrasekaran, "An expert system approach to malfunction diagnosis in chemical plants," *Comput. Chem. Eng.*, vol. 12, no. 1, pp. 27-36, 1988.
[17] H. A. Simon, *Sciences of the Artificial*, 2nd ed. Cambridge, MA: M.I.T. Press, 1981.
[18] J. W. Smith, J. R. Svirbely, C. A. Evans, P. Straum, J. R. Josephson, and M. C. Tanner, "RED: A red-cell antibody identification module," *J. Med. Syst.*, vol. 9, no. 3, pp. 121-138, 1985.

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, somehow 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

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 $U_1, \cdots, U_n$, and m labels, denoted by $L_1, \cdots, L_m$. Each U_i $(i = 1, \cdots, n)$ will be assigned a set of m numbers $p_1(i), \cdots, p_m(i)$ called a labeling distribution:

$$p_1(i) \geq 0, \cdots, p_m(i) \geq 0, \tag{1}$$

$$\sum_{j=1}^{m} p_j(i) = 1. \tag{2}$$

For abbreviation, we let

$$p(i) \triangleq (p_1(i), \cdots, p_m(i)), \quad i = 1, 2, \cdots, n, \tag{3}$$

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_k at the unit U_h influences the label L_j at the unit U_i. If the unit U_h having the label L_k lends a high support to the unit U_i having the label L_j, then $r(i, j; h, k)$ should be large and positive. If constraints are such that the unit U_h having the label L_k means that the label L_j at the unit U_i 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

$$r(i, j; i, j) = \text{a positive constant, for instance } \alpha, \text{ independent of } i, j, \tag{4}$$

$$r(i, j; h, k) \leq \alpha. \tag{5}$$

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

Define the support on the unit U_i having the label L_j with another labeling distribution component $v_j(i)$, $\{U_i, L_j, v_j(i)\}$, from $\{U_h, L_k, p_k(h)\}$ by $r(i, j; h, k)p_k(h)v_j(i)$.

Define the support on $\{U_i, L_j, v_j(i)\}$ from the unit U_h having a label distribution $p(h)$, $\{U_h, p(h)\}$, by $\Sigma_{k=1}^{m} r(i, j; h, k)p_k(h)v_j(i)$.

Define the support on $\{U_i, L_j, v_j(i)\}$ from n labeling distributions, $p(1), \cdots, p(n)$, by $q_j(i; P)v_j(i)$ where

$$q_j(i; P) \triangleq \sum_{h=1}^{n} \sum_{k=1}^{m} r(i, j; h, k)p_k(h), \tag{6}$$

$$p \triangleq [p(1), \cdots, p(n)]. \tag{7}$$

Define the support on $\{U_i, v(i)\}$ from P by $\Sigma_{j=1}^{m} q_j(i; P)v_j(i)$.

Define the support on n labeling distributions, $v(1), \cdots, v(n)$, or $V \triangleq [v(1), \cdots, v(n)]$ from P by $\Sigma_{i=1}^{n} \Sigma_{j=1}^{m} q_j(i; P)v_j(i)$.

For abbreviation, we let

$$q(i; P) \triangleq (q_1(i; P), \cdots, q_m(i; P)),$$
$$q(P) \triangleq [q(1; P), \cdots, q(n; P)]$$

and simply call each of P and V a labeling. Thus the support on the labeling V from the labeling P is represented by the inner product $(q(P), V)$ in the nm-dimensional Euclidean space E_{nm}:

$$(q(P), V) = \sum_{i=1}^{n} (q(i; P), v(i)), \tag{8}$$

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 $p(1), \cdots, p(n)$ is called unambiguous if each of n units is assigned a unique label, that is, for each i, $1 \leq i \leq n$, all $p_j(i)$'s $(j = 1, \cdots, m)$ 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 $p(1), \cdots, p(n)$ comprise a consistent labeling if for various n labeling distributions $v(1), \cdots, v(n)$ there hold the following variational inequalities:

$$(q(i; P), v(i) - p(i)) \leq 0, \quad i = 1, \cdots, n, \tag{9}$$

or the same

$$(q(i; P), v(i)) \leq (q(i; P), p(i)), \quad i = 1, \cdots, n. \tag{10}$$

In other words, P is a consistent labeling if and only if $\forall i$, $i = 1, \cdots, n$, $p(i)$ 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

$$\begin{aligned}(q(P), V) &= \sum_{i=1}^{n} (q(i; P), v(i)) \\ &\leq \sum_{i=1}^{n} (q(i; P), p(i)) \\ &= (q(P), P).\end{aligned} \tag{11}$$

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

$$(q(P), V) \leq (q(P), P), \tag{12}$$

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^n [see (16)].

Let $e_1, \cdots, e_m$ be m standard basis vectors in E_m. Let

$$K_0 = \{e_1, \cdots, e_m\}, \tag{13}$$

$$K = \left\{\sum_{j=1}^{m} u_j e_j : u_j \geq 0, \sum_{j=1}^{m} u_j = 1\right\}, \tag{14}$$

$$K_0^n = \underbrace{K_0 \times, \cdots, \times K_0}_{n \text{ times}}, \tag{15}$$

$$K^n = \underbrace{K \times, \cdots, \times K}_{n \text{ times}}. \tag{16}$$

Then $K(K^n)$ is a linear convex set in $E_m(E_{nm})$ and $K_0(K_0^n)$ the set of vertices of $K(K^n)$. The set K takes a specific name "simplex" in topology and linear programming. It is clear that $q(P)$ $(q(i; P))$ defines a linear transformation: $K^n \to E_{nm}(E_m)$ and $q_j(i; P)$ a linear functional: $K^n \to E_1$. The inner product $(q(P), V)$ defines a bilinear functional: $K^n \times K^n \to E_1$ and the inner product $(q(i; P), v(i))$ a bilinear functional: $K^n \times K \to E_1$.

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

$$\big(q(i; P), v(i)\big) < \big(q(i; P), p(i)\big), \quad i = 1, \cdots, n. \tag{17}$$

In other words, P is a strictly consistent labeling if and only if $\forall i$, $i = 1, \cdots, n$, $p(i)$ 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

$$\big(q(P), V\big) < \big(q(P), P\big). \tag{18}$$

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 $P = [p(1), \cdots, p(n)]$ with $p(i) = (p_1(i), \cdots, p_m(i))$ we need first to consider

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

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 a linear convex set. Let $M_0(i; P)$ be simply the set of vertices which correspond to components of $q(i; P)$ where the component of $q(i; P)$ attains the maximum value:

$$M_0(i; P) \triangleq \Big\{e_j: \big(q(i; P), e_j\big) = \max_{1 \le k \le m} \big(q(i; P), e_k\big)\Big\}. \tag{20}$$

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

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

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

$$M_0(i; P) = \Big\{e_j: q_j(i; P) = \max_{1 \le k \le m} q_k(i; P)\Big\}. \tag{22}$$

and hence

$$M(i; P) = \left\{\sum_{j=1}^{m} u_j e_j: u_j \ge 0, \sum_{j=1}^{m} u_j = 1, \quad u_j = 0 \text{ if } e_j \notin M_0(i; P)\right\}. \tag{23}$$

Since P is a consistent labeling if and only if $\forall i$, $i = 1, \cdots, n$, $p(i)$ 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:

$$p(i) \in M(i; P), \quad i = 1, \cdots, n. \tag{24}$$

Since P is a strictly consistent labeling if and only if $\forall i$, $i = 1, \cdots, n$, $p(i)$ 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:

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

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

Let

$$M_0(P) = M_0(1; P) \times, \cdots, \times M_0(n; P), \tag{26}$$

$$M(P) = M(1; P) \times, \cdots, \times M(n; P). \tag{27}$$

Then we can also characterize a consistent labeling P by:

$$P \in M(P), \tag{28}$$

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

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

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 $\forall V \neq P$, $\|V - P\| \ll$ ($\ll$ means "small enough"), it holds that

$$V \notin Z. \tag{30}$$

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

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

$$M(i; P) \subset M(i; P^0), \quad i = 1, \cdots, n, \tag{31}$$

or briefly

$$M(P) \subset M(P^0), \tag{31$'$}$$

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

For any two labelings P' and P'', it holds that

$$M(i; P') \cap M(i; P'') \subset M\big(i; tP' + (1 - t)P''\big), \quad i = 1, \cdots, n, \tag{32}$$

or briefly

$$M(P') \cap M(P'') \subset M\big(tP' + (1 - t)P''\big), \tag{32$'$}$$

where $0 \le t \le 1$.

To prove (31)–(32) we need only to prove that

$$M_0(i; P) \subset M_0(i; P^0), \quad i = 1, \cdots, n, \tag{33}$$

whenever $\|P - P^0\|$ is small, and

$$M_0(i; P') \cap M_0(i; P'') \subset M_0\big(i; tP' + (1 - t)P''\big), \quad i = 1, \cdots, n, \tag{34}$$

where $0 \le t \le 1$.

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

Suppose $e_j \in M_0(i; P') \cap M_0(i; P'')$. Then $\forall k$ we have

$$q_j(i; P') \ge q_k(i; P'),$$

$$q_j(i; P'') \ge q_k(i; P''),$$

and hence $\forall t$, $0 \le t \le 1$,

$$q_j\big(i; tP' + (1 - t)P''\big) \ge q_k\big(i; tP' + (1 - t)P''\big),$$

since $q_j(i; P)$ is a linear functional w.r.t. P. The last inequality implies $e_j \in M_0(i; tP' + (1 - t)P'')$. This 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 $\|P - P^0\| \ll$ by (31)$'$. Thus $\forall P$, $P \neq P^0$ and $\|P - P^0\| \ll$, 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 $P', P'' \in Z$ and $M(P') \cap M(P'') \neq \phi$, then $\forall t$, $0 < t < 1$, $tP' + (1 - t)P'' \in$

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:

$$\left(q\left(tP' + (1 - t)P''\right), P\right) = \left(q\left(tP' + (1 - t)P''\right), tP' + (1 - t)P''\right),$$

or after manipulations

$$\begin{aligned} 0 &= t^2\left(q(P'), P - P'\right) + (1 - t)^2\left(q(P''), P - P''\right) \\ &\quad + t(1 - t)\left[\left(q(P'), P - P''\right) + \left(q(P''), P - P'\right)\right] \\ &= t(1 - t)\left(q(P' - P''), P' - P''\right), \end{aligned} \tag{35}$$

where both $(q(P'), P - P')$ and $(q(P''), P - P'')$ are equal to zero and canceled because of $P' \in M(P')$, $P'' \in M(P'')$ and $P \in M(P') \cap M(P'')$. From (35) it is clear that $\forall t$, $0 < t < 1$, $tP' + (1 - t)P'' \in Z$ if and only if $(q(P' - P''), P' - P'') = 0$. It is also clear that the condition, $(q(P' - P''), P' - P'') = 0$, is implied by $t_0 P' + (1 - t_0)P'' \in Z$ for some $0 < t_0 < 1$. That completes the proof.

In summary, two consistent labelings can be connected by "a line segment" on which each point is a consistent labeling if there exists a third point on the line segment which is a consistent labeling.

V. A Simplex Algorithm

Similar to linear programming, the previous reasoning first leads to the maximal vertex set, $M_0(P) \subset K_0^n$, and then the maximal point set, $M(P) \subset K^n$, formed by $M_0(P)$, where each $M(i; P)$ comprises a face of the simplex K. If $P \in M(P)$, then P is a consistent labeling. If not, what is the next candidate consistent labeling to choose? Suppose $W(P) \triangleq [w(1; P), \cdots, w(n; P)]$ is the orthogonal projection of P onto $M(P)$. That is,

$$W(P) \in M(P), \ \left\|W(P) - P\right\| = \min_{V \in M(P)} \|V - P\|. \tag{36}$$

It is apparent that $W(P)$ is uniquely determined by P and each $w(i; P)$ is the orthogonal projection of $p(i)$ onto $M(i; P)$, i.e.,

$$w(i; P) \in M(i; P), \ \left\|w(i; P) - p(i)\right\| = \min_{v(i) \in M(i; P)} \left\|v(i) - p(i)\right\|. \tag{36$'$}$$

A consistent labeling P could be characterized as:

$$P = W(P) \quad \text{or} \quad p(i) = w(i; P), \qquad i = 1, \cdots, n. \tag{37}$$

When $P \notin Z$, it seems reasonable to choose $W(P)$ as the next candidate consistent labeling.

Let

$$w_j(i; P) = \begin{cases} p_j(i) + \sum_{k: e_k \notin M_0(i; P)} p_k(i) / \# M_0(i; P), \\ \qquad \text{if } e_j \in M_0(i; P), \\ 0, \qquad \text{otherwise,} \end{cases} \tag{38}$$

$$w(i; P) = \left(w_1(i; P), \cdots, w_m(i; P)\right),$$

$$W(P) = \left[w(1; P), \cdots, w(n; P)\right].$$

Then $w(i; P)$ $(W(P))$ defined by (38) is the orthogonal projection of $p(i)$ (P) onto $M(i; P)$ $(M(P))$. It is easy to see that

$$w_j(i; P) \geq 0, = 0 \qquad \text{if } e_j \notin M_0(i; P),$$

$$\sum_{j=1}^{m} w_j(i; P) = \sum_{j: e_j \in M_0(i; P)} p_j(i) + \sum_{k: e_k \notin M_0(i; P)} p_k(i) = 1.$$

and for any $v(i) \in M(i; P)$

$$\begin{aligned} &\left(v(i), w(i; P) - p(i)\right) \\ &\quad = \sum_{j: e_j \in M_0(i; P)} v_j(i)\left[w_j(i; P) - p_j(i)\right] \\ &\quad = \left(\sum_{j: e_j \in M_0(i; P)} v_j(i)\right)\left(\sum_{k: e_k \notin M_0(i; P)} p_k(i)\right) \Big/ \# M_0(i; P) \\ &\quad = \sum_{k: e_k \notin M_0(i; P)} p_k(i) / \# M_0(i; P), \end{aligned}$$

which is independent of $v(i)$. Therefore, $w(i; P)$ belongs to $M(i; P)$ and comprises the unique orthogonal projection of $p(i)$ onto $M(i; P)$.

Now we are able to summarize the algorithm:

Step 1. Set P^1.
Step 2. Set $k = 1$.
Step 3. Compute $M_0(P^k)$.
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 $q(i; P)$, $i = 1, \cdots, n$.
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^0 is a strictly consistent labeling. Then, when P^k is close to P^0, only one more iteration is needed to reach the goal P^0. That is,

$$P^{k+1} = P^0. \tag{39}$$

Proof: Since P^0 is a strictly consistent labeling. $M(P)$ will consist of a single point P^0, whenever $\|P - P^0\|$ is small, as argued before. Thus, when P^k is close to P^0, it holds that

$$M(P^k) = \{P^0\},$$

which implies that

$$P^{k+1} = W(P^k) = P^0,$$

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

$$M(P^k) \subset M(P^0), \qquad k \geq k_0,$$

which is implied by (31).

According to the algorithm, P^{k+1}, being the orthogonal projection of P^k onto $M(P^k)$, should belong to $M(P^k)$ 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

$$\left\|P - P^0\right\| \geq \left\|W(P^0) - P^0\right\|.$$

Case	Initial Distributions				Algorithm 1 after 25 Iterations				Algorithm 2 after 1 Iteration			
A	0.25	0.25	0.25	0.25	0.27	0.27	0.23	0.23				
	0.25	0.25	0.25	0.25	0.27	0.27	0.23	0.23				
	0.25	0.25	0.25	0.25	0.27	0.27	0.23	0.23				
B	0.5	0	0.5	0	0.99	0	0.01	0	1	0	0	0
	0.5	0	0.5	0	0.99	0	0.01	0	1	0	0	0
	0.5	0	0.5	0	0.99	0	0.01	0	1	0	0	0
C	0.5	0	0.5	0	0.99	0	0.01	0	1	0	0	0
	0.4	0	0.6	0	0.91	0	0.09	0	1	0	0	0
	0.5	0	0.5	0	0.99	0	0.01	0	1	0	0	0
D	0.5	0	0.5	0	1	0	0	0	1	0	0	0
	0.3	0	0.7	0	0.19	0	0.81	0	1	0	0	0
	0.5	0	0.5	0	1	0	0	0	1	0	0	0
E	0.3	0	0.7	0	0.9	0	0.1	0	1	0	0	0
	0.3	0	0.7	0	0.9	0	0.1	0	1	0	0	0
	0.5	0	0.5	0	1	0	0	0	1	0	0	0
F	0.2	0	0.8	0	0.07	0	0.93	0	1	0	0	0
	0.3	0	0.7	0	1	0	0	0	1	0	0	0
	0.5	0	0.5	0	1	0	0	0	1	0	0	0
G	0.3	0.2	0.3	0.2	0.98	0	0.02	0	1	0	0	0
	0.3	0.2	0.3	0.2	0.98	0	0.02	0	1	0	0	0
	0.3	0.2	0.3	0.2	0.98	0	0.02	0	1	0	0	0
H	0.3	0.2	0.3	0.2	1	0	0	0	1	0	0	0
	0.25	0.25	0.25	0.25	1	0	0	0	1	0	0	0
	0.2	0.2	0.4	0.2	0.11	0	0.89	0	0	0	1	0
I	0.5	0	0.5	0	1	0	0	0	1	0	0	0
	0.02	0	0.98	0	0	0	1	0	0	0	1	0
	0.5	0	0.5	0	1	0	0	0	1	0	0	0

Fig. 1. Experimental result of the line labeling.

Especially when $k \geq k_0$, P^{k+1} belongs to $M(P^0)$, it holds that

$$\|P^{k+1} - P^0\| \geq \|W(P^0) - P^0\|. \tag{40}$$

which concludes $W(P^0) = P^0$ or $P^0 \in Z$ because of $\|P^{k+1} - P^0\| \to 0$.

Q.E.D.

VII. Comparison to the Rosenfeld *et al.* [3] Consistent Labeling Definition

Using the same notation as in Section II, the Rosenfeld *et al.* relaxation labeling update scheme is as follows:

$$p_j(i) := \frac{p_j(i)\left[1 + q_j(i; P)\right]}{\sum_{k=1}^{m} p_k(i)\left[1 + q_k(i; P)\right]},$$
$$j = 1, \cdots, m; \quad i = 1, \cdots, n. \tag{41}$$

When $|r(i, j; h, k)| <<$, their assumption, $|q_j(i; P)| < 1$, will be satisfied. A labeling P is consistent in Rosenfeld *et al.*'s sense if P is a fixed point of (41). An essential condition of a consistent labeling in the Rosenfeld *et al.* sense is 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 is Rosenfeld *et al.*'s consistent labeling. Suppose P is a Hummel and Zucker consistent labeling. Then for each i, $i = 1, \cdots, n$, $p(i)$ belongs to $M(i; P)$, which means that for each $p_j(i) > 0$, $q_j(i; P) = \max_{1 \leq k \leq m} q_k(i; P)$, a constant independent of j. It completes the proof.

Since Hummel and Zucker's consistent labeling set is nonempty, Rosenfeld *et al.*'s consistent labeling set is nonempty as well.

VIII. Experimental Results and Summary

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 λ on unit a_i with label λ' on unit a_j, $r_{ij}(\lambda, \lambda')$, is related to the function $r(i, \lambda; j, \lambda')$ in this paper as:

$$r(i, \lambda; j, \lambda') = d_{ij} \cdot r_{ij}(\lambda, \lambda')$$

where the d's are constant coefficients. Then, the function $q_i^{(k)}(\lambda)$ which is the change in $p_i^{(k)}(\lambda)$ in the kth iteration, where $q_i^{(k)}(\lambda)$ are the notation used in [3], is same as the support function $q_\lambda(i; P^k)$ in the new algorithm. Using the same values for $r_{ij}(\lambda, \lambda')$ 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 labels L_i $(i = 1, \cdots, 4)$, the set of four line labels $\{+, -, \leftarrow, \rightarrow\}$ 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: $\{(\rightarrow, \rightarrow), (\rightarrow, -), (-, \rightarrow), (\leftarrow, \leftarrow), (\leftarrow, +), (+, \leftarrow)\}$. 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

$$\begin{matrix} 0.5 & 0.5 & 0 & 0 \\ 0.5 & 0.5 & 0 & 0 \\ 0.5 & 0.5 & 0 & 0 \end{matrix}$$

Case	Initial Distributions			Algorithm 2 after 25 Iterations			Algorithm 2 after 1 Iteration		
A	0.33	0.33	0.33	0.94	0.03	0.03	1	0	0
	0.33	0.33	0.33	0.94	0.03	0.03	1	0	0
	0.33	0.33	0.33	0.94	0.03	0.03	1	0	0
B	0.4	0.3	0.3	1	0	0	1	0	0
	0.33	0.33	0.33	0	1	0	1	0	0
	0.33	0.33	0.33	0	0	1	1	0	0
C	0.5	0.25	0.25	1	0	0	1	0	0
	0.4	0.3	0.3	0.06	0.94	0	1	0	0
	0.4	0.3	0.3	0.06	0	0.94	1	0	0
D	0.6	0.2	0.2	1	0	0	1	0	0
	0.5	0.25	0.25	0.99	0.01	0	1	0	0
	0.5	0.25	0.25	0.99	0	0.01	1	0	0
E	0.33	0.33	0.33	0	1	0	1	0	0
	0.3	0.3	0.4	0	0	1	1	0	0
	0.33	0.33	0.33	1	0	0	1	0	0
F	0.3	0.3	0.4	0.75	0	0.25	1	0	0
	0.3	0.4	0.3	0.75	0.25	0	1	0	0
	0.33	0.33	0.33	0.06	0.47	0.47	1	0	0
G	0.3	0.3	0.4	1	0	0	1[a]	0	0
	0.25	0.5	0.25	0	1	0	0	1	0
	0.33	0.33	0.33	0	0	1	0	0	1
H	0.58	0.21	0.21	1	0	0	1	0	0
	0.33	0.33	0.33	0	1	0	0	1	0
	0.33	0.33	0.33	0	0	1	0	0	1
I	0.74	0.12	0.12	1	0	0	1	0	0
	0.4	0.3	0.3	0	1	0	0	1	0
	0.4	0.3	0.3	0	0	1	0	0	1
J	0.33	0.33	0.33	0	1	0	0*	1	0
	0.29	0.29	0.42	0	0	1	0	0	1
	0.33	0.33	0.33	1	0	0	1	0	0

[a]Takes 2 iterations to reach the final state.

Fig. 2. Experimental result of the junction labeling.

and the second iteration in the same case gives

$$\begin{matrix} 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0.5 & 0.5 \end{matrix}$$

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

$$\begin{matrix} 0.27 & 0.27 & 0.23 & 0.23 \\ 0.27 & 0.27 & 0.23 & 0.23 \\ 0.27 & 0.27 & 0.23 & 0.23 \end{matrix}$$

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 H both algorithms give a less 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 an 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.

Based on fully exploiting the linearity of the variational inequality and linear convexity of consistent labeling search space, a simplex-like algorithm is developed for the relaxation labeling process. Its effectiveness is thus far substantiated by both theory and experiments.

Acknowledgment

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

References

[1] R. Hummel and S. Zucker, "On the foundations of relaxation labeling processes," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-5, p. 267, 1983.

[2] D. Kinderlehrer and G. Stampacchia, *An Introduction to Variational Inequalities and Their Applications.* New York: Academic, 1980.

[3] A. Rosenfeld, R. Hummel, and S. Zucker, "Scene labeling by relaxation operations," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-6, p. 420, 1976.

[4] D. L. Waltz, "Generating semantic descriptions from drawings of scenes with shadows," MIT Tech. Rep. AI 271, Nov. 1972.