Multiple Subclass Pattern Recognition: A Maximin Correlation Approach

Hadar I. Avi-Itzhak, Jan A. Van Mieghem, and Leonardo Rub

Abstract—**This paper addresses a correlation based nearest neighbor pattern recognition problem where each class is given as a collection of subclass templates. The recognition is performed in two stages. In the first stage the class is determined. Templates for this stage are created using the subclass templates. Assignment into subclasses occurs in the second stage. This two stage approach may be used to accelerate template matching. In particular, the second stage may be omitted when only the class needs to be determined.**

We present a method for optimal aggregation of subclass templates into class templates. For each class, the new template is optimal in that it maximizes the worst case (i.e., minimum) correlation with its subclass templates. An algorithm which solves this maximin optimization problem is presented and its correctness is proved. In addition, test results are provided, indicating that the algorithm's execution time is polynomial in the number of subclass templates.

We show tight bounds on the maximin correlation. The bounds are functions only of the number of original subclass templates and the minimum element in their correlation matrix.

The algorithm is demonstrated on a multifont optical character recognition problem.

Index Items—**Pattern recognition, nearest neighbor, template matching, correlation, maximin, minimax, clustering, multifont optical character recognition.**

I. Introduction

PATTERN recognition often involves the assignment of a pattern, represented by a vector of measured features, to one of N classes using a library of pre-classified vectors which are known as template vectors. In optical character recognition, for example, a class could correspond to one letter of the alphabet. A commonly used approach, known as nearest-neighbor recognition [2], [5], [6] assigns the input vector to the class whose template vector is most "similar" to it. Different functions are described in the literature to measure similarity [5], [6], [10], [25], [26]. In particular, the correlation operator (normalized scalar product) is used in many image analysis applications [1], [11], [15], [20], [23] where invariance to intensity scaling is desirable. With this measure, the input and the template having the largest correlation are nearest neighbors. The value of this correlation may be used to indicate a confidence level in the classification decision.

Manuscript received November 23, 1993; revised August 25, 1994.

H. Avi-Itzhak is with Canon Research Center America, 4009 Miranda Ave., Palo Alto, CA 94304.

J.A. Van Mieghem is with the Graduate School of Business, Stanford University, CA 94305-5015.

L. Rub is with Octel Communications Corp., 1001 Murphy Ranch Rd., Milpitas, CA 95035.

IEEECS Log Number P95037.

With multiple-subclass pattern recognition some or all of the classes are represented by more than one template, each representing a different subclass. The major discrepancies between the different templates of a class are not due to random effects such as noise but, rather, each template represents a different subclass. To continue the example of optical character recognition, the class representing a letter could be composed of several subclasses, each corresponding to a different style or font. Font variations due to different printer makes can also be interpreted as different subclasses.

Directly implementing the nearest neighbor algorithm based on all of the original templates becomes computationally intensive as the number of subclasses increases. It is therefore advantageous to approach the problem in two stages [8], [16], [21], [24], [27]. The class is determined first, subsequently allowing a more efficient recognition of the subclass. The first stage is performed without using the subclass information. Therefore, for each class, we are interested in aggregating the templates of its subclasses into a single template representative of the class.[1] This paper addresses the problem of the construction of the optimal aggregate template, using correlation as a similarity measure.

Our approach is based on the following premise: the aforementioned aggregation results in a loss of information which translates into a risk of increased probability of error, i.e., incorrect classification. The incurred risk may vary from subclass to subclass. For each class we would like to choose a representative template which insures a minimal worst case risk over all of its subclasses. In other words, we wish to find the template which minimizes the maximum risk. Depending on the particular probability distributions, the probability of error is assumed to be monotonically decreasing with the correlation with the nearest-neighbor template. For example, for additive Gaussian noise this relationship is exponential. Therefore, the desired representative template vector maximizes the minimum correlation with the templates it replaces.

An optical character recognition example is used to illustrate this concept. Fig. 1 depicts 11 sentences, each printed in one of 11 common English fonts. The printout was scanned at 400 dots per inch to create a 3000 by 3000 matrix of gray scale values in the range of 0 to 255, which was then segmented into 339 isolated character matrices. Each character matrix was scaled to have a second moment of 10, and translated to have its center of mass at the center of a 50 by 50

[1] A class may be represented by more than one template. In this case, the class is partitioned into clusters of subclass templates, which are separately aggregated.

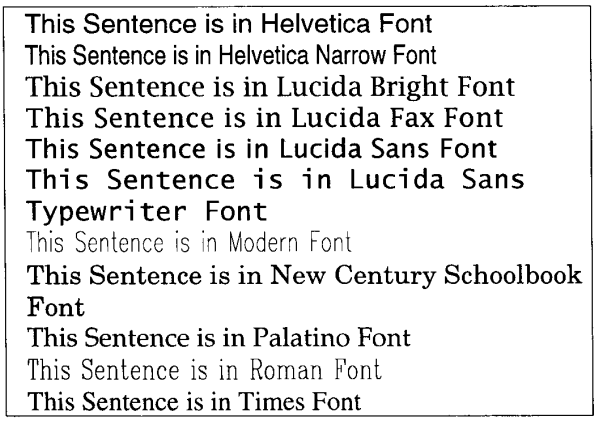


Fig. 1. Printout used for 11 to one example.

matrix. The columns of each resultant matrix were concatenated to form a 2500 element vector which represents the normalized optical measurements of its respective printed character.

In addition, a complete set of the English alphabet was printed 11 times, each time using a different one of the 11 fonts, and the same segmentation and normalization procedure was used to generate a set of 11 templates for each character. Next, for each character, the maximin template was computed from its 11 template vectors. These maximin templates were then used for correlation based nearest neighbor recognition of the characters of Fig. 1. All 339 characters were recognized correctly. For comparison, the recognition was repeated, this time with each maximin template vector being replaced by the arithmetic mean of the 11 template vectors which were used to compute it. This time, only 329 out of the 339 characters were recognized correctly.

In the next section, we state the formulation of the maximin aggregation problem. In Section III, we present a fast algorithm that computes the solution of the maximin problem. Section IV establishes a series of properties of the solution which enable us to show the correctness of the algorithm. Also, upper and lower bounds on the maximin correlation are derived. In Section V, we show the equivalence of the maximin correlation problem to a quadratic program. We discuss the computational performance of the algorithm in Section VI. Tests show that the algorithm is polynomial in the number of templates. As a demonstration, the algorithm is applied to multifont optical character recognition in Section VII. Finally, we conclude in Section VIII.

II. Problem Formulation

Since the nearest neighbor is determined based on the correlation operator, without loss of generality all template and input vectors may be scaled to unit norm. Now the correlation is reduced to an inner product, and the domain of vectors is restricted to the surface of the unit hypersphere. To gain further insight we can also use the fact that the correlations are monotonically decreasing with distances or angles. Then, by visualizing the surface of the unit hypersphere as a plane (Fig. 2), the problem becomes more intuitive. Fig. 3 uses this visualization to illustrate a three-class multiple subclass example where 15 templates are aggregated into three representative templates. Note that all the points in the figure are in reality on the surface of the hypersphere.

Following normalization, we can formulate the problem for a single class containing m subclasses as follows:

Given a set $S = \{t_1, t_2, t_3, \cdots, t_m\}$ of m template column vectors in $\mathbf{R}^n$ (all vectors are assumed to be column vectors in $\mathbf{R}^n$ unless noted otherwise), we will make use of the following assumptions:

- A1. The vectors in S are linearly independent.
- A2. For all $t_s \in S$ we have $\|t_s\| = 1$.
- A3. For all $t_{s1}, t_{s2} \in S$ we have $t_{s1} \cdot t_{s2} \geq 0$.

In practical settings n is large such that the probability of having two linearly dependent vectors is negligible which is assumed in A1. Assumption A2 represents the scaling of the templates due to normalization. Because t_{s_1}, t_{s_2} represent subclass templates of the same class, one can reasonably expect them to have a non-negative correlation as in A3.

We wish to find a vector $t_M \in \mathbf{R}^n$ which solves the following optimization problem

$$t_M = \arg\max_{\|t\|=1} \left\{ \min_{t_s \in S} t \cdot t_s \right\}. \tag{1}$$

We will denote the optimal objective value by ρ_M and the objective function being maximized by

$$F(t) = \min_{t_s \in S} t \cdot t_s. \tag{2}$$

We note that the unit hypersphere is a compact set on which $F(t)$ is bounded. Therefore, t_M and ρ_M exist, and the problem is well posed. Furthermore, we will show that the solution t_M is unique.

It will be convenient to introduce the following quantities: T, the $n \times m$ matrix with template t_i as i^{th} column vector; $C = T'T$, the $m \times m$ symmetric correlation matrix ; and e, a column vector of all ones ("summer vector"). Transposes are denoted by primes. The minimum (maximum) element of a vector or matrix A will be denoted by min A (max A). Given sets A and B, the set $\{x : x \in A \text{ and } x \notin B\}$ will be denoted by $A - B$.

III. Solution of the Maximin Correlation Problem

In this section, we state the main result of the paper, the solution to the maximin correlation problem. The mathematical infrastructure developed in sections IV.A and IV.B will lead to the proof of the main result in section IV.C.

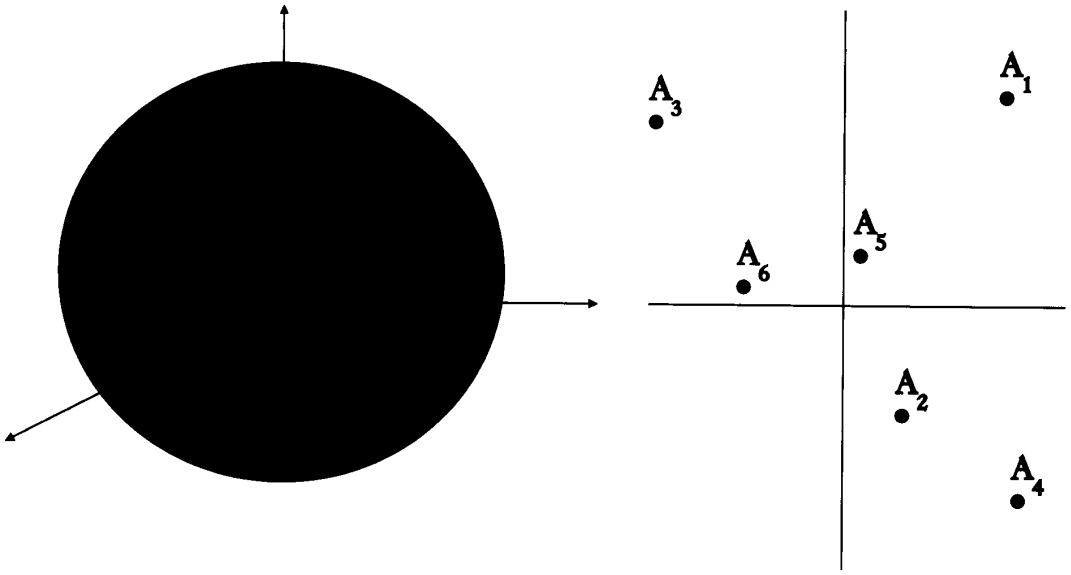


Fig. 2. Looking at the hypersphere's surface as a plane.

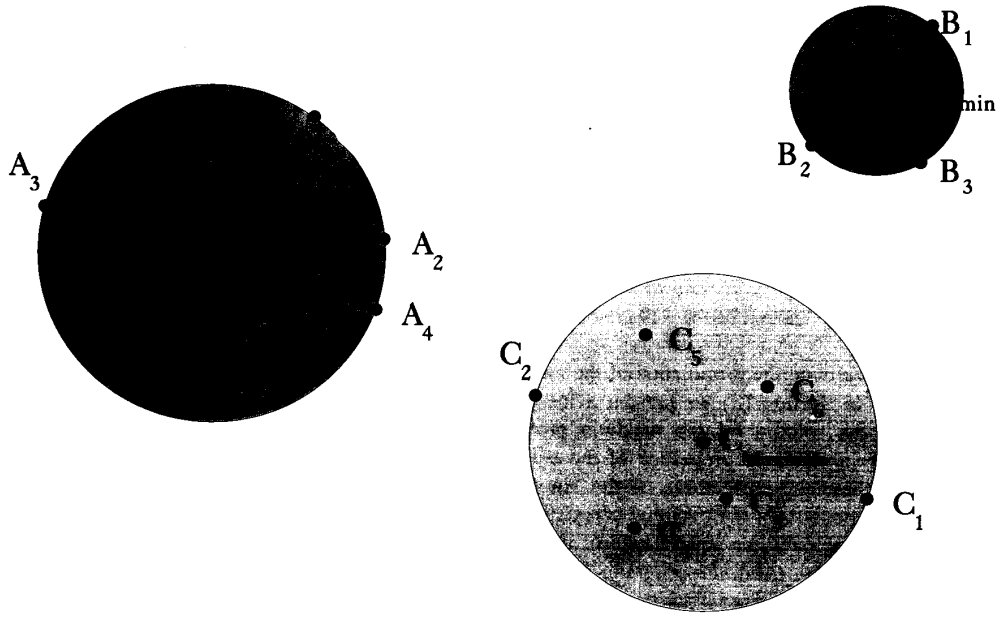


Fig. 3. Conceptual illustration of multiple subclass maximin correlation.

Theorem 1(Main Result): *The following algorithm computes t_M and ρ_M in a finite number of iterations.*

Step 1: Let $t = \frac{Te}{\|Te\|}$, where e is the m-dimensional summer vector. Let $\Sigma = arg \min_{t_s \in S} t \cdot t_s$.

Step 2: a) Let $E^*(\Sigma) = \frac{T_\Sigma C_\Sigma^{-1} e}{\sqrt{e' C_\Sigma^{-1} e}}$, where T_Σ is the $n \times k$ matrix whose columns are the template vectors contained in the set Σ, $C_\Sigma = T_\Sigma' T_\Sigma$, and e is the k-dimensional summer vector. If $t = E^*(\Sigma)$, go to step 4.

b) Let $\lambda_s = \frac{t \cdot (t_s - t_\sigma)}{(t_s - t_\sigma)(t - E^*(\Sigma))}$, where t_σ is any template in Σ, and $t_s \in S - \Sigma$.

If any λ_s satisfies $0 < \lambda_s < 1$, then let $\lambda_{\min} = \min_{0<\lambda_s<1} \lambda_s$, otherwise let $\lambda_{\min} = 1$.

Let $t = \dfrac{\lambda_{\min} E^*(\Sigma) + (1-\lambda_{\min})t}{\left\| \lambda_{\min} E^*(\Sigma) + (1-\lambda_{\min})t \right\|}$.

c) Let $\Sigma = arg \min_{t_s \in S} t \cdot t_s$.

If $\lambda_{\min} = 1$ (i.e. if $t = E^*(\Sigma)$) then go to step 3, otherwise go to 2a.

Step 3: If $w^* \geq 0$, where $w^* = C_\Sigma^{-1} e$, go to step 4.
Otherwise, let i be any index that satisfies $w_i^* = \min w^*$, let $\Sigma = \Sigma - \{t_i\}$, and go to step 2.

Step 4: The solution is $t_M = t$ and $\rho_M = \min T' t_M$. Terminate the algorithm.

IV. Properties of the Solution

A. Equicorrelated Points Set $E(\Sigma)$

We will show that there exist at least two template vectors t_i and t_j such that $t_M \cdot t_i = t_M \cdot t_j$. We say that t_M is *equicorrelated* to t_i and t_j. In order to facilitate the exposition we introduce the following definitions.

Definition: Given a subset $\Sigma \subseteq S$ of $k > 1$ template vectors, the set $E(\Sigma)$ of their *equicorrelated points* is

$$E(\Sigma) = \left\{ t \in \mathbb{R}^n : \forall t_i, t_j \in \Sigma \text{ with } i \neq j \Rightarrow t \cdot t_i = t \cdot t_j \right\}. \quad (3)$$

If $k = 1$, then $E(\Sigma) = \mathbb{R}^n$.

Let T_Σ be the $n \times k$ matrix whose columns are the template vectors in Σ. For any $t \in E(\Sigma)$ there exists a scalar ρ such that

$$T_\Sigma' t = \rho e. \quad (4)$$

Because T_Σ has rank k (assumption A1) and $t = 0$ is in $E(\Sigma)$, it follows that $E(\Sigma)$ is a linear subspace of dimension $n - k + 1$. Hence, t can be expressed as $t = t_\| + t_\perp$ where $t_\|$ is in the span of Σ and $t_\perp$ is in its null space. In other words, there exists a k-element vector w such that $t_\| = T_\Sigma w$, and $T_\Sigma' t_\perp = 0$. From (4) it follows that

$$E(\Sigma) = \left\{ t \in \mathbb{R}^n : t = \rho T_\Sigma C_\Sigma^{-1} e + t_\perp \text{ where } \rho \in R \text{ and } T_\Sigma' t_\perp = 0 \right\} \quad (5)$$

where $C_\Sigma = T_\Sigma' T_\Sigma$. Note that the parameter ρ is the scalar product of t with the template vectors in Σ.

Definition: Given a subset $\Sigma \subseteq S$ of k template vectors, its *optimally equicorrelated point* $E^*(\Sigma)$ is defined as

$$E^*(\Sigma) = arg \max_{t \in E(\Sigma), \|t\|=1} \{ t \cdot t_\sigma \text{ for any } t_\sigma \in \Sigma \}. \quad (6)$$

Because the feasible set $\{ t \in \mathbb{R}^n : t \in E(\Sigma), \|t\| = 1 \}$ is nonempty (since $n - k + 1 > 0$) and compact, and $t \cdot t_\sigma$ is bounded on this set, a maximum exists. The following lemma shows that this maximum is unique.

Lemma 1: *The optimally equicorrelated point of Σ is*

$$E^*(\Sigma) = \rho_\Sigma^* T_\Sigma C_\Sigma^{-1} e, \quad (7)$$

where $\rho_\Sigma^* = \left(e' C_\Sigma^{-1} e \right)^{-1/2}$.

Proof: Using (5), (6) is equivalent to

$$\rho_\Sigma^* = arg \max_{\rho \in R} \rho \qquad \text{subject to } \rho^2 e' C_\Sigma^{-1} e + \|t_\perp\|^2 = 1. \quad (8)$$

Clearly the maximum is obtained when $t_\perp = 0$, from which Lemma 1 follows. □

Fig. 4(a) shows the equicorrelated points set and the optimally equicorrelated point for a simple case of two template vectors. Fig. 4(b) is the planar visualization of Fig. 4(a).

B. Necessary and Sufficient Conditions

The following two lemmas are needed to derive the necessary and sufficient conditions for the solution of the maximin problem.

Lemma 2: (Extension of Minimum to Neighborhood) *Given a vector $t_0 \in \mathbb{R}^n$ and a subset of template vectors $\Sigma = \{ t_i \in S: F(t_0) = t_0 \cdot t_i \}$ where $F(t)$ is defined in (2), there exists an $\varepsilon > 0$ such that*

$$\forall t \in \mathbb{R}^n : \|t - t_0\| < \varepsilon \Rightarrow F(t) = \min_{t_\sigma \in \Sigma} t \cdot t_\sigma. \quad (9)$$

Note that by definition $F(t) = \min_{t_s \in S} t \cdot t_s$, so the claim of the lemma is that in an ε-neighborhood of t_0 the minimization need only be carried out over the subset Σ. In other words, at a small enough distance from t_0, the template vectors with the smallest correlation are still in the subset of template vectors with the smallest correlation to t_0.

Proof: If $\Sigma = S$, the lemma is trivial. Therefore, we will consider only proper subsets in which case we let $\Delta = \min_{t_s \in S - \Sigma} \{ t_0 \cdot t_s \} - F(t_0)$, where $S - \Sigma = \{ t_s \in S \text{ and } t_s \notin \Sigma \}$.

From the assumptions of the lemma, it is clear that $\Delta > 0$. Setting $\varepsilon = \Delta/3$, we have that

$$\forall t \in \mathbb{R}^n : \|t - t_0\| < \varepsilon \Rightarrow |t \cdot t_i - t_0 \cdot t_i| = |(t - t_0) \cdot t_i| \leq \|t - t_0\| < \varepsilon = \Delta/3,$$

for any $t_i \in S$, which implies that

$$t_0 \cdot t_i - \Delta/3 < t \cdot t_i < t_0 \cdot t_i + \Delta/3.$$

If $t_i \in \Sigma$, the right inequality implies that $t \cdot t_i < F(t_0) + \Delta/3$. Otherwise, the left inequality yields $t \cdot t_i > \min_{t_s \in S - \Sigma} \{ t_0 \cdot t_s \} - \Delta/3 = F(t_0) + 2\Delta/3$. Therefore, we must have that $F(t) = \min_{t_\sigma \in \Sigma} \{ t \cdot t_\sigma \}$. □

We will now show that the vector t_M which solves the maximin problem (1) must belong to the equicorrelated point set of at least two of the template vectors and that the remaining (if any) template vectors will have a higher correlation with t_M.

Lemma 3: (Weak Version of Necessary Conditions) *If t_M is a solution of the maximin problem (1), then there exists a subset $\Sigma \subseteq S$ of at least two template vectors such that:*

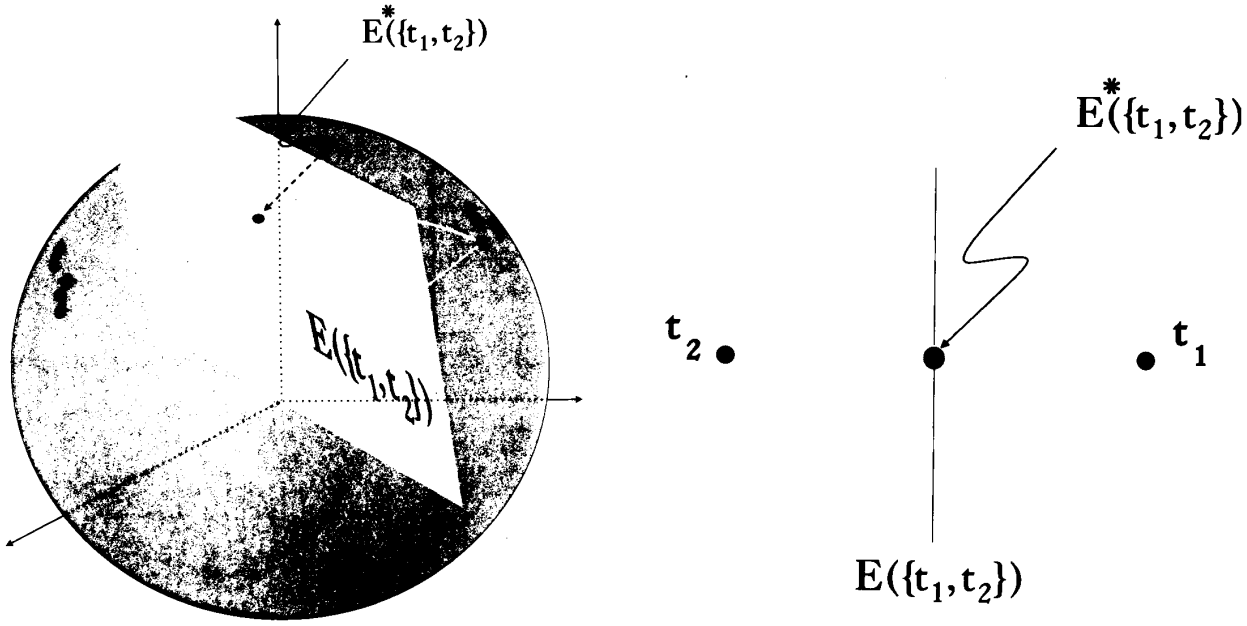


Fig. 4. Equicorrelated points (left) with planar visualization (right).

$$t_M \in E(\Sigma), \tag{10}$$

and the correlation of the remaining (if any) template vectors with t_M is strictly greater than ρ_M:

$$\forall t_i \in S-\Sigma \text{ and } \forall t_\sigma \in \Sigma: \rho_M = t_M \cdot t_\sigma < t_M \cdot t_i. \tag{11}$$

Proof (by contradiction): Assume there is a unique index i^* such that $\rho_M = F(t_M) = t_M \cdot t_{i^*}$. According to Lemma 2, there exists an ε-neighborhood of t_M in which for all t, $F(t) = t \cdot t_{i^*}$. Let $t_\delta = \alpha(t_M + \delta t_{i^*})$, where $\alpha = \left(1+\delta^2+2\delta t_M \cdot t_{i^*}\right)^{-1/2}$. For a small $\delta > 0$, the vector t_δ will be in this ε-neighborhood, so $F(t_\delta) = t_\delta \cdot t_{i^*} = \alpha(t_M \cdot t_{i^*} + \delta)$. Unless in the trivial case where $S = \Sigma = \{t_{i^*}\}$, we have that $0 \le \rho_M = t_M \cdot t_{i^*} < 1$, in which case it is easy to show that $F(t_\delta) > \rho_M$. Together with the fact that $\|t_\delta\| = 1$, this contradicts the optimality of t_M which proves Lemma 3. □

We are now in the position to prove a stronger and more useful version of Lemma 3 which states that the vector t_M which solves the maximin problem (1) must be the *optimally* equicorrelated point of a subset $\Sigma \subseteq S$.

Theorem 2: (Necessary Conditions) *If t_M is a solution of the maximin problem (1), then there exists a subset $\Sigma \subseteq S$ of at least two template vectors such that:*

$$t_M = E^*(\Sigma), \tag{12}$$

and the correlation of the remaining (if any) template vectors with t_M is strictly greater than ρ_M.

Proof (by contradiction): Assume that there is no subset Σ such that $t_M = E^*(\Sigma)$. We know from Lemma 3 that there exists a subset Σ such that $t_M \in E(\Sigma)$ and we know from Lemma 1 that $E^*(\Sigma)$ exists. Analogous to the proof of Lemma 3, let $t_\delta = \alpha[t_M + \delta E^*(\Sigma)]$, where $\alpha = \left[1+\delta^2+2\delta t_M \cdot E^*(\Sigma)\right]^{-1/2}$. Using Lemma 2, we have that for small $\delta > 0$, $F(t_\delta) = t_\delta \cdot t_\sigma = \alpha(t_M \cdot t_\sigma + \delta\rho^*_\Sigma)$, where $t_\sigma \in \Sigma$ and ρ^*_Σ is defined in Lemma 1. Because $t_M \in E(\Sigma)$, $t_M \neq E^*(\Sigma)$, and $t_\sigma \in \Sigma$, the definition of $E^*(\Sigma)$ implies that $t_M \cdot t_\sigma < E^*(\Sigma) \cdot t_\sigma$, and also $t_M \cdot E^*(\Sigma) < 1$ (because $t_M \neq E^*(\Sigma)$). Using these strict inequalities, it is easy to show that $F(t_\delta) > F(t_M)$, which contradicts the optimality of t_M. □

In the example shown in Fig. 5, the necessary conditions are fulfilled with $\Sigma = \{t_1, t_2\}$. In this planar visualization, t_3 is inside the circle, illustrating the fact $E^*(\{t_1,t_2\}) \cdot t_1 < E^*(\{t_1,t_2\}) \cdot t_3$ and $E^*(\{t_1,t_2\}) \cdot t_2 < E^*(\{t_1,t_2\}) \cdot t_3$.

Corollary: *The maximin solution t_M is unique.*

Proof: Since $F(t)$ is concave, and $\|t\| \le 1$ is a convex set, the solution of the maximin problem is a convex set in $\mathbf{R}^n$. Therefore, the solution set has either a single or an uncountable number of points. From Theorem 2 we know that ρ_M can be achieved by at most $2^m - m - 1$ vectors. Thus t_M must be unique. □

Lemma 4: *Given $F_\Sigma(t) = \min t' T_\nu$ and a unit vector $x \in$ span(Σ), then, for any unit-length y in the ε-neighborhood of x, we can find a unit-length $z \in$ span(Σ), also in the neighborhood, such that $F_\Sigma(z) \ge F_\Sigma(y)$.*

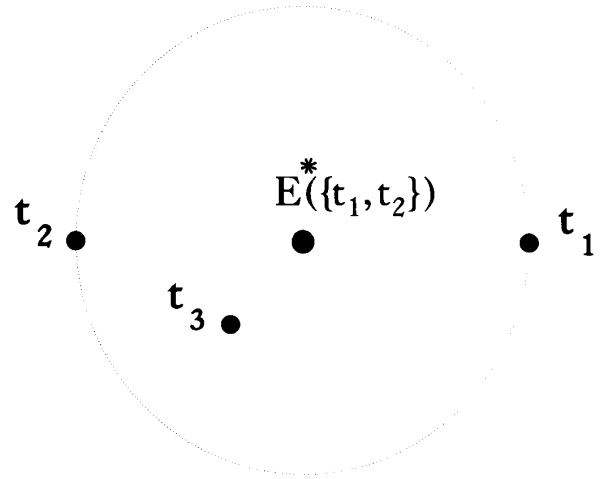


Fig. 5. Necessary conditions.

Proof: y can be written as $y = \frac{x+\delta}{\|x+\delta\|}$, and δ can be represented as $\delta = \delta_{\|} + \delta_{\perp}$ where $\delta_{\perp} \cdot x = 0$. Given such a y, we consider $z = \frac{x+\delta_{\|}}{\|x+\delta_{\|}\|}$.

$$\|z-x\|^2 = 2 - 2z \cdot x = 2 - 2\frac{1 + x \cdot \delta_{\|}}{\sqrt{1 + 2x \cdot \delta_{\|} + \|\delta_{\|}\|^2}}$$

$$\leq 2 - 2\frac{1 + x \cdot \delta_{\|}}{\sqrt{1 + 2x \cdot \delta_{\|} + \|\delta\|^2}} = \|y-x\|^2 < \varepsilon^2$$

Thus, the vector z which we selected is in the ε-neighborhood of x. For any $t_\sigma \in \Sigma$,

$$t_\sigma \cdot z = \frac{t_\sigma \cdot x + t_\sigma \cdot \delta_{\|}}{\sqrt{1+2x\cdot\delta_{\|}+\|\delta_{\|}\|^2}} \geq \frac{t_\sigma \cdot x + t_\sigma \cdot \delta_{\|}}{\sqrt{1+2x\cdot\delta_{\|}+\|\delta\|^2}} = t_\sigma \cdot y.$$

Thus, $F_\Sigma(z) \geq F_\Sigma(y)$. □

The above lemma implies that in order to optimize $F_\Sigma(t)$ in the ε-neighborhood of $x \in \Sigma$, it is sufficient to optimize in *span*(Σ).

Theorem 3: (Necessary and sufficient conditions) *Given $t^* = E^*(\Sigma)$, which satisfies the necessary conditions of Theorem 2, then $t^* = t_M$ if and only if $C_\Sigma^{-1} e \geq 0$.*

Proof: Since $F(t)$ is concave, in order to show that $t^* = t_M$, it is sufficient to show that given an ε-neighborhood of t^*, we cannot find a t in this neighborhood such that $F(t) > F(t^*)$.

Using Lemma 2, we can choose an ε such that for any t, if $\|t - t^*\| < \varepsilon$, then $F(t) = F_\Sigma(t) = \min t' T_\Sigma$.

From Lemma 4 we know that in order to find a t in the ε-neighborhood of t^* such that $F(t) > F(t^*)$, we only have to consider $t \in$ *span*(Σ). Therefore, there exists a unique w such that $t = T_\Sigma w$, and a unique w^* such that $t^* = T_\Sigma w^*$. Since $\|t\| = 1$, it follows that :

$$w' C_\Sigma w = 1. \tag{13}$$

We will now examine the changes in $F(t)$ as a function of w. Let $\delta_w = w - w^*$, where $\|\delta_w\|$ can be chosen small enough to insure that $\|t - t^*\| < \varepsilon$. For each $t_\sigma \in \Sigma$ we define a function $f_\sigma = t_\sigma' T_\Sigma w$. When w changes by δ_w, f_σ changes by $t_\sigma' T_\Sigma \delta_w$. Noting that $t_\sigma' T_\Sigma$ is the σth row of C_Σ, the changes in all f_σs can be represented as $C_\Sigma \delta_w$. In order for $F(t)$ to increase as a result of the change δ_w we must have :

$$b > 0 \text{ where } b = C_\Sigma \delta_w, \tag{14}$$

because when $w = w^*$, all f_σ are equal and must *all* increase in order for $\min f_\sigma$ to increase.

The vector δ_w cannot be completely arbitrary. Any w must satisfy (13). Substituting $w = w^* + \delta_w$ in (13) we get $2\delta_w' C_\Sigma w^* = -\delta_w' C_\Sigma \delta_w$. Since C_Σ is positive definite, we obtain

$$\delta_w' C_\Sigma w^* < 0.$$

Substituting $\delta_w = C_\Sigma^{-1} b$ yields

$$b' w^* < 0. \tag{15}$$

Using (14) and (15), we can prove the two implications stated in the theorem:

From (14), we must have $b > 0$ to have an increase in $F(t)$. If in addition $w^* \geq 0$, then (15) does not hold, and we must have $t^* = t_M$.

On the other hand, suppose that $t^* = t_M$, and at least one component of w^* is negative. Then it is possible to find a vector $b > 0$ such that (15) holds. We can then determine the corresponding $\delta_w = C_\Sigma^{-1} b$, since C_Σ is non singular. Thus, we get the contradictory statement $t^* \neq t_M$, and we must have $w^* \geq 0$. □

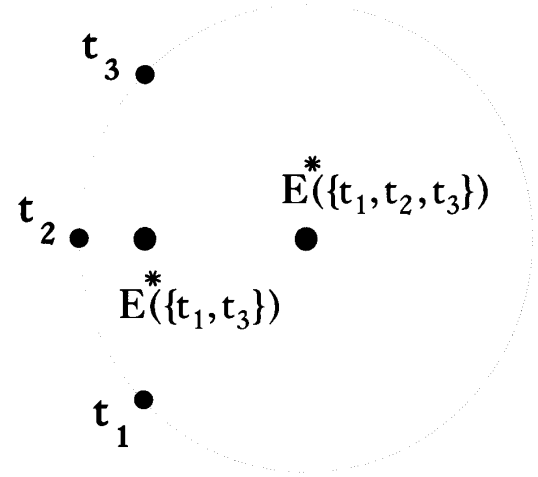


Fig. 6. Sufficient conditions.

In Fig. 6, both $E^*(\{t_1, t_3\})$ and $E^*(\{t_1, t_2, t_3\})$ satisfy the necessary conditions. In the planar visualization shown in the figure, $E^*(\{t_1, t_3\})$ appears inside the convex hull of t_1 and t_3.

This represents the fact that in reality it is inside their convex cone, thus satisfying the sufficient conditions. $E^*(\{t_1, t_2, t_3\})$, however, appears outside the convex hull of t_1, t_2, and t_3 in this two dimensional representation, and is therefore outside their convex cone in higher dimensions, thus violating the sufficient conditions. Since the other two candidates, $E^*(\{t_1, t_2\})$ and $E^*(\{t_2, t_3\})$ (not shown in Fig. 6) do not satisfy the necessary conditions, we must have $t_M = E^*(\{t_1, t_3\})$.

We are now in a position to show that the Algorithm of Section III is correct.

C. Explanation and Proof of the Algorithm

The algorithm is based on the following approach. The initial value of t is used to find an improved t which is guaranteed to satisfy the weak necessary conditions (Lemma 3), i.e., $t \in E(\Sigma)$, and $t \cdot t_\sigma < t \cdot t_s$, for $t_\sigma \in \Sigma$, and $t_s \in S - \Sigma$. The repeated iterations of step 2 are attempts to strengthen the conditions to those of Theorem 2. A "line" search (actually a great arc on the unit hypersphere) is performed between t and $E^*(\Sigma)$, the optimally equicorrelated point of Σ. Ascent of $F(t)$ is guaranteed as long as the vector satisfies the conditions of Lemma 3 (see Lemma 5 below). These conditions are satisfied up to

$$\frac{\lambda_{\min} E^*(\Sigma) + (1-\lambda_{\min}) t}{\left\| \lambda_{\min} E^*(\Sigma) + (1-\lambda_{\min}) t \right\|} \quad \text{(see Lemma 6 below).}$$

If this point is reached, then t becomes equicorrelated to a larger subset of S, for which the conditions of the Lemma are satisfied. The iteration can then be repeated with a line search to the optimally equicorrelated point of this new subset of S.

Since there are m vectors in S and, at each iteration, at least one vector is added to Σ, then there can be no more than m iterations before reaching step 3, and achieving the necessary conditions of Theorem 2. If the sufficient conditions of Theorem 3 are also satisfied, we have the optimal t. Otherwise, a new set Σ such that t still satisfies the conditions of Lemma 3 is obtained by removing the appropriate template from Σ (see Lemma 7). Each time step 3 is reached, the subset Σ is different from any previous subsets because of the strict ascent in $F(t)$ during steps 2 and 3. The number of different subsets of S is 2^m. Thus, the algorithm must terminate in less than $m2^m$ iterations. Extensive tests have shown empirically that the actual number of iterations is approximately linear in m, as documented in Section VI.

Lemma 5: *Let $t_0 \in E(\Sigma)$, and $t(\lambda) = \frac{\lambda E^*(\Sigma) + (1-\lambda) t_0}{\left\| \lambda E^*(\Sigma) + (1-\lambda) t_0 \right\|}$. If $t(\lambda)$ satisfies the conditions of Lemma 3, then $F(t(\lambda))$ strictly increases with λ in the subinterval of $0 \le \lambda \le 1$, where the conditions are satisfied.*

Proof: From (5), we can represent t as :

$$t(\lambda) = \rho(\lambda) T_\Sigma C_\Sigma^{-1} e + t_\perp(\lambda) \text{ where } \rho(\lambda)^2 e' C_\Sigma^{-1} e + \left\| t_\perp(\lambda) \right\|^2 = 1.$$

Note that $\rho(\lambda)$ is the correlation of $t(\lambda)$ with the template vectors in Σ. Since $E^*(\Sigma)$ has no orthogonal component, $\left\| t_\perp(\lambda) \right\|$ is a strictly decreasing function of λ. Thus $\rho(\lambda)$ is a strictly increasing function of λ. As long as the conditions of Lemma 3 hold, $F(t(\lambda)) = \rho(\lambda)$, so that $F(t)$ is strictly increasing in λ. □

The following lemma mentions $\lambda_{\min}$, which is defined in step 2b as $\lambda_{\min} = \min_{0<\lambda_s<1} \lambda_s$. For each t_s, λ_s determines the point in the line search where t_s is considered for inclusion in Σ. Thus, λ_s is obtained by solving $t(\lambda_s) \cdot t_s = t(\lambda_s) \cdot t_\sigma$, where t_σ is any template in Σ.

Lemma 6: *Consider t_0 and $t(\lambda)$ of Lemma 5. Then, the conditions of Lemma 3 are satisfied in the interval $0 \le \lambda \le \lambda_{\min}$.*

Proof: Suppose that for λ_1, where $0 < \lambda_1 < \lambda_{\min}$, the conditions of Lemma 3 are not satisfied. Then, there is a $t_s \in S - \Sigma$ and a $t_\sigma \in \Sigma$ such that $t(\lambda_1) \cdot t_s < t(\lambda_1) \cdot t_\sigma$. Since $t(0) \cdot t_s > t(0) \cdot t_\sigma$, then, by the intermediate value theorem[2] there must be a λ_0 such that $0 < \lambda_0 < \lambda_1$ and $t(\lambda_0) \cdot t_s = t(\lambda_0) \cdot t_\sigma$. Thus, we have $\lambda_s = \lambda_0 < \lambda_{\min}$, contradicting the definition of $\lambda_{\min}$. Therefore, the conditions of Lemma 3 must be satisfied in the interval $0 \le \lambda \le \lambda_{\min}$. □

Lemma 7: *Consider the set of templates Σ at the beginning of step 3, and the template t_i defined in step 3. If*

$$t(\lambda) = \frac{\lambda E^*(\Sigma - \{t_i\}) + (1-\lambda) E^*(\Sigma)}{\left\| \lambda E^*(\Sigma - \{t_i\}) + (1-\lambda) E^*(\Sigma) \right\|},$$

then

$$t(\lambda) \cdot t_i > t(\lambda) \cdot t_j,$$

where $0 < \lambda \le 1$, $i \ne j$, and $t_j \in \Sigma$.

Proof: Since $E^*(\Sigma) \in E(\Sigma - \{t_i\})$, and $E^*(\Sigma) \ne E^*(\Sigma - \{t_i\})$, then from the proof of Lemma 5 all $t(\lambda) \cdot t_j$, for $j \ne i$, increase equally with λ. Thus, for $j \ne i$ we have $b_j = \alpha$, where b_j represents the change in $t(\lambda) \cdot t_j$ and $\alpha > 0$. According to (15) we must have $b'w^* < 0$, leading to :

$$b_i w_i^* + \alpha \sum_{j \ne i} w_j^* < 0.$$

Noting that $w_i^* < 0$ from its definition in step 3 in the algorithm :

$$b_i > -\frac{\alpha \sum_{j \ne i} w_j^*}{w_i^*}. \tag{16}$$

Since C_Σ^{-1} is positive definite we have $e' C_\Sigma^{-1} e > 0$. Thus, $e' w^* > 0$ and

$$-\frac{\sum_{j \ne i} w_j^*}{w_i^*} > 1. \tag{17}$$

Substituting (17) into (16) yields $b_i > \alpha$, completing the proof of the lemma. □

[2] $t(\lambda) \cdot t_\sigma$ and $t(\lambda) \cdot t_s$ are continuous functions of λ.

D. Bounds to the Maximin Solution

Theorem 4: *The maximin solution has the following upper bound*

$$\rho_M \le \sqrt{\frac{1+\min C}{2}}, \tag{18}$$

where $\min C$ *is the smallest element in* C.

Proof: If $\Sigma \subseteq S$, then for any t:

$$\min t'T \le \min t'T_\Sigma.$$

Therefore,

$$\rho_M \le \max_{\|t\|=1}\{\min t'T_\Sigma\}.$$

Letting $\Sigma = \{t_p, t_q\}$, we get from Theorem 2

$$\max_{\|t\|=1}\{\min t'T_\Sigma\} = \rho_\Sigma^* = \sqrt{\frac{1+t_p \cdot t_q}{2}}.$$

So for any $t_p, t_q \in S$, we have

$$\rho_M \le \sqrt{\frac{1+t_p \cdot t_q}{2}}.$$

Choosing t_p and t_q so that $t_p \cdot t_q = \min C$, we get the tightest version of the above bound:

$$\rho_m \le \sqrt{\frac{1+\min C}{2}}$$

□

Definition (Pairwise Equicorrelated Template Set): A set of vectors Π is a pairwise equicorrelated template set if there exists a ρ $(0 \le \rho < 1)$ such that for all $t_i, t_j \in \Pi$

$$t_i \cdot t_j = \begin{cases} \rho & : \ i \ne j \\ 1 & : \ i = j \end{cases}$$

In other words, a pairwise-equicorrelated template set is a set of vectors on the unit hypersphere such that the correlations of all of the pairs of vectors are equal. The following lemma will introduce pairwise-equicorrelated template sets for the determination of a lower bound to the maximin solution.

Lemma 8 (Lower Bound to ρ_Σ^*) *Consider any set* Σ *of* k *template vectors with minimum pairwise correlation* $\rho = \min C_\Sigma$. *Let* C_Π *be the* $k \times k$ *correlation matrix with off-diagonal elements equal to* ρ, *and let* Π *be any pairwise equicorrelated template set that generates* C_Π. *If* ρ_Σ^* *is the maximin solution of* Σ, *then*

$$\rho_\Sigma^* \ge \rho_\Pi^*, \text{ where } \rho_\Pi^* = \sqrt{\frac{1}{k} + \left(1 - \frac{1}{k}\right)\rho}.$$

Proof: Because $C_\Pi = (1-\rho)I + \rho ee'$, it has the following eigenvalues: $\lambda_1 = \cdots = \lambda_{k-1} = 1-\rho$, and $\lambda_k = 1-\rho+k\rho$. Hence, C_Π is a positive definite, C_Π^{-1} exists and has a similar structure: $\alpha I + \beta ee'$. We can solve $\frac{1}{\lambda_1} = \alpha$ and $\frac{1}{\lambda_k} = \alpha + k\beta$ to yield

$$C_\Pi^{-1} = \frac{1}{\lambda_1} I + \frac{1}{k}\left(\frac{1}{\lambda_k} - \frac{1}{\lambda_1}\right) ee' = \frac{1}{1-\rho} I + \frac{1}{k}\left[\frac{1}{1+(k-1)\rho} - \frac{1}{1-\rho}\right] ee'. \tag{19}$$

Since C_Π is a symmetric positive definite matrix, we can always find a square root matrix T_Π such that $T_\Pi' T_\Pi = C_\Pi$. The columns of T_Π make up a pairwise equicorrelated template set. Therefore, given ρ, we can always find a pairwise equicorrelated template set that generates C_Π.

Define $x = C_\Sigma^{-1} e$ and $y = C_\Pi^{-1} e$. Since ρ_Σ^* is the maximin solution to Σ, then from Theorem 3 we have $x \ge 0$. Let $\Delta_C = C_\Sigma - C_\Pi$ ($\Delta_C > 0$ from the definition of C_Π) and consider the equations

$$C_\Sigma x = (C_\Pi + \Delta_C)x = e, \tag{20}$$

$$C_\Pi y = e. \tag{21}$$

Subtracting (21) from (20) and pre-multiplying by e' yields

$$[1+(k-1)\rho](e'x - e'y) + e'\Delta_C x = 0.$$

Since $x \ge 0$ and $\Delta_C \ge 0$, we must have $e'\Delta_C x \ge 0$, as well as

$$e'x \le e'y \Rightarrow e'C_\Sigma^{-1}e \le e'C_\Pi^{-1}e \Rightarrow \frac{1}{\sqrt{e'C_\Sigma^{-1}e}} \ge \frac{1}{\sqrt{e'C_\Pi^{-1}e}}.$$

Thus, $\rho_\Sigma^* \ge \rho_\Pi^*$.

We can now use (19) to obtain ρ_Π^*:

$$\rho_\Pi^* = \frac{1}{\sqrt{e'C_\Pi^{-1}e}} = \sqrt{\frac{\lambda_k}{k}} = \sqrt{\frac{1}{k} + \left(1 - \frac{1}{k}\right)\rho}. \tag{22}$$

□

We are now in a position to state a tight lower bound to the maximin correlation for an arbitrary template set S.

Theorem 5 (Lower Bound to the Maximin Solution) *Given any set* S *with* m *templates, its maximin solution has the following lower bound*

$$\rho_M \ge \sqrt{\frac{1}{m} + \left(1 - \frac{1}{m}\right)\min C}. \tag{23}$$

Proof: From Theorem 2 we know that $\rho_M = \rho_\Sigma^*$, where $\Sigma \subseteq S$ has k templates, $2 \le k \le m$. Denoting $\rho = \min C_\Sigma$, we have from Lemma 8:

$$\rho_M \ge \sqrt{\frac{1}{k} + \left(1 - \frac{1}{k}\right)\rho} \ge \sqrt{\frac{1}{m} + \left(1 - \frac{1}{m}\right)\rho}.$$

Since $\rho \ge \min C$,

$$\rho_M \ge \sqrt{\frac{1}{m} + \left(1 - \frac{1}{m}\right)\min C}.$$

□

Corollary: *The above lower bound is tight.*

Proof: We will show that for the set Π of m pairwise equicorrelated templates, $\rho_M = \rho^*_\Pi$. In other words, we will show that Π achieves the lower bound.

Suppose that $\rho_M = \rho^*_\Sigma$, where $\Sigma \subset \Pi$ (for convenience, relabel the t_i such that $\Sigma = \{t_1, t_2, \ldots, t_k\}$). $E^*(\Sigma) = T_\Sigma w$, where w can be obtained using (19):

$$w = \frac{C_\Sigma^{-1} e}{\sqrt{e' C_\Sigma^{-1} e}} = \frac{e}{\sqrt{\lambda_k k}}, \text{ where } \lambda_k = 1 - \rho + k\rho.$$

We can now compare $E^*(\Sigma) \cdot t_m$ with ρ^*_Π:

$$E^*(\Sigma) \cdot t_m = \sum_{i=1}^{k} w_i t_i \cdot t_m = \frac{1}{\sqrt{\lambda_k k}} \sum_{i=1}^{k} t_i \cdot t_m$$
$$= \frac{k\rho}{\sqrt{\lambda_k k}} = \rho \sqrt{\frac{k}{\lambda_k}} < \sqrt{\rho} < \rho^*_\Pi,$$

where we have made use of the fact that $\frac{\lambda_k}{k} > \rho$. Thus, we have a contradiction with the optimality assumption on $E^*(\Sigma)$. Hence, the optimum is achieved by $E^*(\Pi)$ and Π achieves the lower bound. □

V. Transformation into a Quadratic Programming Formulation

In this section we will show the equivalence of the maximin correlation problem to a quadratic programming problem with linear constraints. We call a function $f(t)$ *positive homogeneous* if for any $t \in \mathbf{R}^n$ and any scalar $k > 0$, we have $f(kt) = kf(t)$.

Lemma 9 (Homogeneous Duality) *Given positive scalars α and β and positive homogeneous functions $g(t)$ and $h(t)$. If $g(t) > 0$ for any $t \neq 0$, and if $\arg\max_{g(t)=\alpha} h(t)$ and $\arg\min_{h(t)=\beta} g(t)$ exist, then*

1) $\left(\max_{g(t)=\alpha} h(t)\right)\left(\min_{h(t)=\beta} g(t)\right) = \alpha\beta$.
2) $\arg\max_{g(t)=\alpha} h(t) = k \arg\min_{h(t)=\beta} g(t)$ *for some scalar k.*

Proof: For any scalars $k_1, k_2 > 0$, we have that

$$\max_{g(t)=\alpha} h(t) = \alpha \max_{g(t)=\alpha} \frac{h(t)}{g(t)} = \alpha \max_{g(k_1 t)=k_1\alpha} \frac{h(k_1 t)}{g(k_1 t)} = \alpha \max_{v \neq o} \frac{h(v)}{g(v)}, \tag{24}$$

and similarly,

$$\min_{h(t)=\beta} g(t) = \beta \min_{h(t)=\beta} \frac{g(t)}{h(t)} = \beta \min_{h(k_2 t)=k_2\beta} \frac{g(k_2 t)}{h(k_2 t)} = \beta \min_{v \neq o} \frac{g(v)}{h(v)}. \tag{25}$$

This, together with $\max_{v \neq 0} \frac{h(v)}{g(v)} = \left(\min_{v \neq 0} \frac{g(v)}{h(v)}\right)^{-1}$ proves the lemma. □

Theorem 6: *The Maximin Correlation Problem is equivalent to a Quadratic Programming Problem:*

$$\max_{\|t\|=1} \left\{ \min_{i \in \{1,2,\cdots,m\}} t \cdot t_i \right\} = \left(\min_{t \cdot t_i \geq 1 \text{ for } 1 \leq i \leq m} \|t\| \right)^{-1}, \tag{26}$$

and the vector which optimizes the left hand side is a normalized version of the vector which optimizes the right hand side.

Proof: Let $h(t) = \min_{i \in \{1,2,\cdots,m\}} t \cdot t_i$ and $g(t) = \|t\|$. Clearly, these functions satisfy the assumptions of Lemma 9. Since $\min_{\min(t \cdot t_i)=1} \|t\| = \min_{t \cdot t_i \geq 1} \|t\|$, the theorem is directly implied by the lemma. □

As a result of Theorem 6, the maximin correlation problem can also be attacked by solving the quadratic programming problem:

$$\min v'v$$
$$\textit{Subject to: } T'v \geq e. \tag{27}$$

The solution to the maximin correlation problem is given by scaling the solution to its "dual" quadratic programming problem:

$$t_M = \frac{v_{\min}}{\|v_{\min}\|} \quad \text{and} \quad \rho_M = \frac{1}{\|v_{\min}\|}. \tag{28}$$

The quadratic programming problem can be solved using a simplex-type method [3], [4], an active set method [7], [17], or an interior-point method [14]. In particular, each of the steps in an active set method can be viewed as a counterpart to each of the steps of the algorithm described in Theorem 1. However, contrary to the proposed maximin correlation algorithm (Theorem 1), a general quadratic programming algorithm does not exploit the specific structure of the problem and will, therefore, not be as efficient.

In general, the first phase of a linear program is used to compute a feasible point for initialization of the active set method. We can use our geometrical insight to efficiently determine an initial condition for the quadratic program. If we denote $v_c = \sum_{i=1}^{m} t_i$, then

$$v_0 = \frac{v_c}{\min_{i \in \{1,2,\cdots,m\}} v_c \cdot t_i} \tag{29}$$

will satisfy the constraints and will always exist because of assumptions A1 and A3.

VI. Computational Performance of the Algorithm

A. Preliminary Analysis

In Section IV, we showed that the algorithm generates t_M in a finite number of iterations. In this section, we will describe an efficient implementation of each iteration and evaluate the overall performance of the algorithm.

There are two parameters that can be considered in order to determine problem size. The first one is the dimension of the vectors, n, and the second one is the number of template vectors, m. Since n only affects the number of computations re-

quired by the inner products, we will focus on the behavior of the algorithm as a function of m.

There are two computationally intensive tasks in each iteration. The first one is the computation of $E^*(\Sigma)$, which requires the generation of $C_\Sigma^{-1}e$. Thus, if Σ has k templates, the above computation in general requires $O(k^3)$ operations. The second computationally intensive task involves the "line" searches needed to update t in step 2b. These searches are performed in $O(n(m-k))$ operations. For a fixed ratio $\frac{n}{m}$, the above translates to $O(m(m-k))$ operations. Therefore, out of the two tasks, the computation of $E^*(\Sigma)$ determines the order of the algorithm. Fortunately, the structure of the algorithm can be exploited to compute C_Σ^{-1} in $O(k^2)$ operations.

B. Efficient Computation of $E^*(\Sigma)$

With every iteration of the algorithm, the set Σ is altered by either adding or removing one template vector.[3] Assume that after the p^{th} iteration, a template vector t_i is included in Σ. In this case :

$$\Sigma_{p+1} = \Sigma_p \cup \{t_i\}, \text{ and } C_{\Sigma_{p+1}} = \begin{bmatrix} C_{\Sigma_p} & T'_{\Sigma_p} t_i \\ t_i' T_{\Sigma_p} & 1 \end{bmatrix}$$

If we define $u = C_{\Sigma_p}^{-1}\left(T'_{\Sigma_p} t_i\right)$, then we can use the Frobenius-Schur inversion formula [9] to compute

$$C_{\Sigma_{p+1}}^{-1} = \frac{1}{\Delta}\begin{bmatrix} \Delta C_{\Sigma_p}^{-1} + uu' & -u \\ -u' & 1 \end{bmatrix}$$

where $\Delta = 1 - u' C_{\Sigma_p} u$. Since $C_{\Sigma_p}^{-1}$ is known, the computation of $C_{\Sigma_{p+1}}^{-1}$ requires only $O(k^2)$ operations.

Similarly, assume that after the p^{th} iteration, a template vector is removed from Σ. Without loss of generality, we can assume that this vector is represented by the last row and column of C_{Σ_p}. Again, we can compute $C_{\Sigma_{p+1}}^{-1}$ efficiently using $C_{\Sigma_{p+1}}^{-1} = C_{inv} - \frac{uu'}{\Delta}$, where Δ is a scalar, and C_{inv} as well as u are defined by the following block matrix representation of $C_{\Sigma_p}^{-1}$:

$$C_{\Sigma_p}^{-1} = \begin{bmatrix} C_{inv} & u \\ u' & \Delta \end{bmatrix}$$

C. Performance Evaluation

In order to evaluate the performance of the algorithm, matrices T with uniformly distributed elements between 0 and 1 were generated, the columns were scaled to unit norm, and the resulting matrices were processed by the algorithm. The number of floating point operations (FLOPS) were determined in a manner identical to MATLAB [18]. For each value of m (number of templates), 10,000 cases were generated using a number of dimensions n, which we arbitrarily set at $n = 1.5m$. Varying m from 10 to 500, the number of iterations can be approximated as $1.2m^{0.7}$, and the number of FLOPS can be approximated as $3.3m^{2.9}$. The exponent of m in the FLOPS is reasonable given that we have a sublinear number of iterations, and each iteration requires approximately $O(m^2)$ FLOPS.

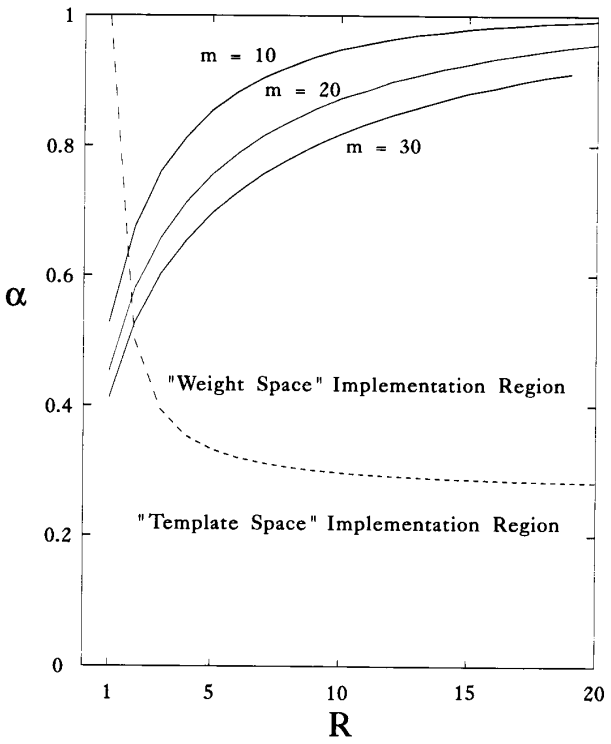


Fig. 7. α vs. R curves for m = 10, 20, and 30 (solid lines). Implementation trade-off curve (dashed line).

The algorithm described in Theorem 1 performs the optimization in the *template space*. Since any template can have the representation $t = Tw$, the optimization can be performed in the *weight space* in terms of w. Instead of computing inner products $t \cdot t_s$ in step 2b, the weight space implementation requires the computation of $w'Cw_s$, where $t = Tw$, and $t_s = Tw_s$. Thus, the weight space implementation has the advantage that each iteration of step 2b is performed in about $4m(m-k)$ operations, rather than $4n(m-k)$, where $n \geq m$ (assumption A1). On the other hand, the weight space implementation requires the computation of all the elements of C, while the template space implementation only requires the elements of C needed to compute $E^*(\Sigma)$ in step 2a. The nature of this trade-off can be further understood with the following analysis.

Test runs indicate that step 3 is rarely performed. Thus, at the k^{th} iteration there are approximately k elements in Σ. Let p denote the total number of iterations executed. The total number of operations in step 2b with the template space approach is :

$$\sum_{k=1}^{p} 4n(m-k) \approx 4nmp - 2np^2.$$

[3] In the exceptionally rare case that more than one template is added to Σ in one iteration, the update procedure to be described is repeated until C_Σ^{-1} is obtained.

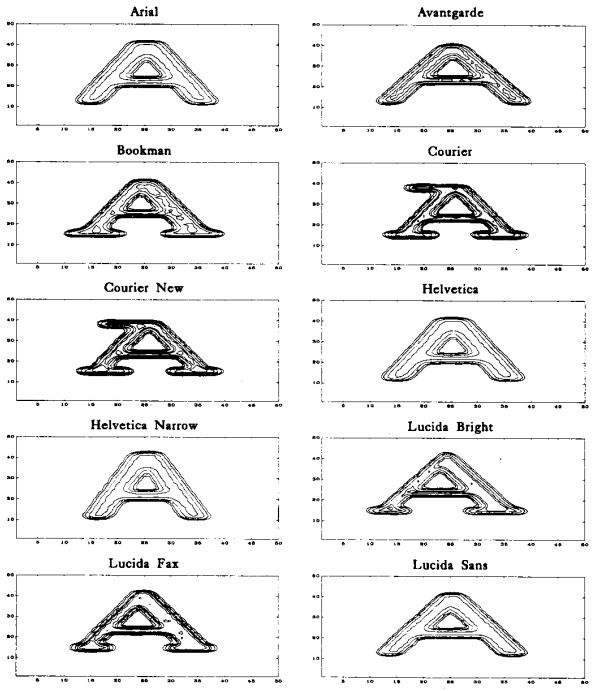


Fig. 8(a). Templates of original subclasses.

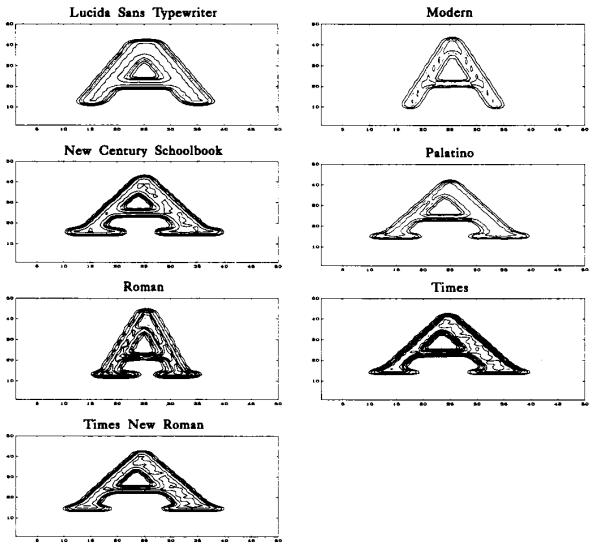


Fig. 8(b). Templates of original subclasses.

The computation of the elements of C needed to find $E^*(\Sigma)$ requires np^2 operations. Thus, the combined number of operations for the template space implementation is

$$4nmp - np^2. \tag{30}$$

Similarly, an approximate expression for the combined number of operations for the weight space implementation is

$$4m^2p - 2mp^2 + nm^2. \tag{31}$$

Using (30), (31), and the representation $p = \alpha m$, and $n = Rm$, we get the following "break-even" curve

$$\alpha_0 = \begin{cases} \frac{2(R-1)-\sqrt{3R^2-6R+4}}{R-2} & : \ R \neq 2 \\ \frac{1}{2} & : \ R = 2 \end{cases} \tag{32}$$

Thus, for any R, the corresponding α must exceed α_0 in order for the weight space implementation to be more efficient than the template space implementation. The decision on a specific implementation must be based on an expectation of the value of α as a function of R. The solid lines in Fig. 7 illustrate the behavior of the mean value α as a function of R for m = 10, 20, and 30 uniformly distributed templates (10,000 template sets were processed for each value of R and m). The dashed line plots (32). The break-even R increases monotonically with m, and reaches $R \approx 2$ when m = 30. Therefore, the weight space implementation will likely be the efficient choice for small values of m, while the opposite will be true for larger values of m.

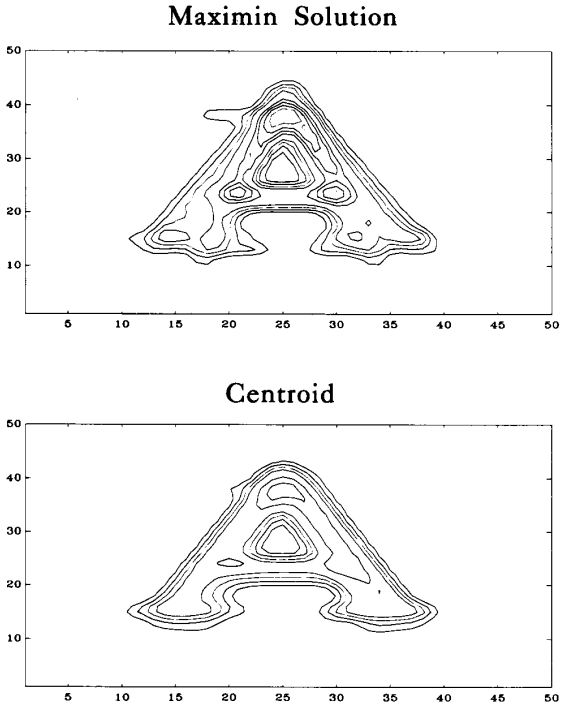


Fig. 9. Aggregate Templates.

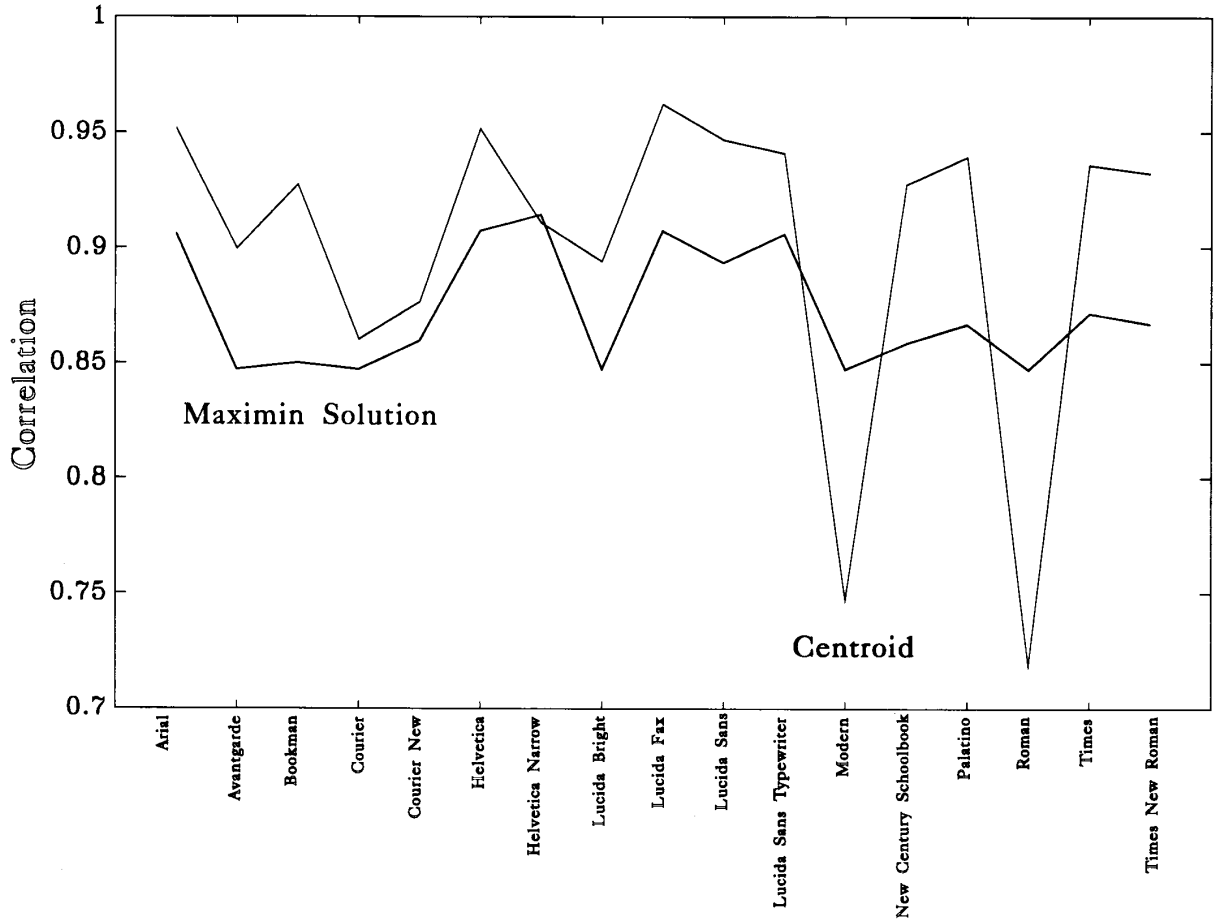


Fig. 10. Correlations of aggregate templates with subclass templates.

VII. OCR Example

A. Multifont Aggregation

In Fig. 8, contour maps of the letter "A" as printed in each of 17 fonts are shown. The data was preprocessed as explained in Section I. The maximin and centroid templates are depicted in Fig. 9. Fig. 10 illustrates the levels of correlations of each of the 17 original templates with the "A" template for both maximin and centroid aggregation. Note that the "Modern" and "Roman" fonts, which, in some ways, might be considered as "outliers" for this collection of fonts, achieve very poor performance with the centroid template, while the maximin aggregate template results in satisfactory correlations with all 17 fonts. In a system designed to recognize these 17 fonts with centroid aggregation, the letter "A" would require two templates to achieve an accuracy equal to or better than a system with one "A" template using maximin aggregation. Alternately, for the same number of aggregate templates, the maximin system would out-perform the centroid system.

B. Recognition Demonstration

Multifont optical character recognition was used to demonstrate maximin correlation template aggregation in the following experiment. For the demonstration, 12 out of the 17 available fonts were used to create maximin aggregate templates. For characters where one aggregate template resulted in misclassification, the original templates were split into separate clusters and each was aggregated separately, resulting in multiple aggregate templates per character. In this fashion, a total of 1128 (94 classes $\times$ 12 subclasses) templates were replaced by 192 aggregate templates, resulting in an average of approximately two aggregate templates per character. Next, the aggregate templates were used to recognize approximately 300 000 isolated characters having point sizes of 10, 12, and 14 and a uniform mixture of the 12 commonly used fonts. The data was generated on one QMS-810 PostScript laser printer. No errors were incurred. The same test using OmniPage Professional Version 2.0, a popular OCR software package supplied by Caere Corp., resulted in a recognition (i.e., excluding segmentation errors) accuracy of 99.44%. For comparison, a benchmark test of OCR performance with current proprietary commercial algorithms was reported by Jenkins et al [12], and the best recognition accuracies that were obtained with comparable data (*i.e.*, high quality) of three point sizes (10, 12, and 14) and three fonts (Courier, Helvetica, and Times-Roman) ranges from 99.21% to 99.95%. Published literature reports

accuracies in the order of 99% to 99.9% for single-font recognition and worse for multi-font [13], [19], [22].

VIII. Conclusion

In this paper we have presented an algorithm which generates templates for the first stage of a two-stage correlation based nearest neighbor pattern recognition implementation. This implementation can be used to accelerate the recognition process or to provide a recognition scheme when each class is defined as a collection of subclass templates. In an optical character recognition example, each letter (class) was defined in terms of a set of templates, each one corresponding to a different font (subclass). Aggregate templates were constructed for each letter, which enabled classification almost six times faster than direct template matching.

We formulated a maximin criterion for optimal template aggregation. In order to minimize the maximum risk of incorrect recognition in the first stage, an aggregate template must maximize the worst case correlation with the original templates. Theoretical foundations were presented, which lead to the main result, the algorithm which solves the maximin optimization problem.

Additionally, the following tight bounds on the maximin correlation ρ_M were derived:

$$\sqrt{\frac{1}{m}+\left(1-\frac{1}{m}\right)\min C} \le \rho_M \le \sqrt{\frac{1+\min C}{2}},$$

where C is the correlation matrix of the subclass templates, and m is the number of original templates. The performance of the algorithm was discussed and the number of iterations and FLOPS were experimentally determined to be of order $O(m^{0.7})$ and $O(m^{2.9})$, respectively, where m represents the number of original templates. It was also shown how the maximin problem can be formulated as a quadratic program. It is known that quadratic programs can be solved in polynomial time which corroborates our experimental performance results.

Finally, the algorithm was demonstrated on a multifont optical character recognition problem and its performance was shown to compare favorably with other algorithms published in the literature.

Acknowledgments

The authors would like to thank Benjamin Avi-Itzhak, Thanh Diep, and Uri Rothblum for their helpful suggestions. Hadar Avi-Itzhak would like to thank the United States Air Force for its fellowship funding.

References

[1] L. Capodiferro, R. Cusani, G. Jacovitti, and M. Vascotto, "Correlation-based technique for shift, scale, and rotation independent object classification," *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing*, Apr. 1987.

[2] T.M. Cover and P.E. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Information Theory*, vol. 13, Jan. 1967.

[3] R.W. Cottle and G.B. Dantzig, "Complementary pivot theory of mathematical programming," *Linear Algebra Applications*, vol 1, pp. 103-105, 1968.

[4] G.B. Dantzig, *Linear Programming and Extensions*, Princeton Univ. Press, 1993.

[5] R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis*. John Wiley and Sons, New York, 1973.

[6] K. Fukunaga and T.E. Flick, "An optimal global nearest neighbor metric," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 6, May 1984.

[7] P.E. Gill, W. Murray, and M.H. Wright, *Practical Optimization*, Academic Press, United Kingdom, 1981.

[8] A. Goshtasby, S.H. Gage, and J.F. Bartholic, "A two-stage cross-correlation approach to template matching," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 6, no. 3, pp. 374-378, May 1984.

[9] M.S. Grewal and A.P. Andrews, *Kalman Filtering : Theory and Practice*, Prentice Hall, 1993.

[10] O. Hoessjer and M. Mettiji, "Robust multiple classification of known signals in additive noise," *IEEE Trans. Information Theory*, vol. 39, no. 2, March 1993.

[11] S. Huang, "Criteria of template matching and image sensor noise analysis," *Proc. IEEE CS Workshop Computer Vision*, pp. 315-317, Miami Beach, Fla., Nov. 1987.

[12] F. Jenkins, J. Kanai, and T.A. Nartker, "Using ideal images to establish a baseline of OCR performance," *Symp. Document Analysis and Information Retrieval*, Las Vegas, Nev., 1993.

[13] S. Kahan, T. Pavlidis, and H.S. Baird, "On the recognition of printed characters of any font and size," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 9, no. 2, pp. 274-288, 1987.

[14] S. Kumar, *Recent Developments in Mathematical Programming*, Gordon and Breach Science Publishers, 1991.

[15] J.C. Lee and E. Milios, "Matching range images of human faces," *Proc. IEEE CS Third Int'l Conf. Computer Vision*, pp. 722-726, 1990.

[16] K. Li, M. Ferdousi, M. Chen, and T.T. Nguyen, "Image matching with multiple templates," *Proc. CS Workshop Computer Vision and Pattern Recognition Conf.*, pp. 610-613, Miami Beach, Fla., June 1986.

[17] D. Luenberger, *Linear and Nonlinear Programming*, Addison-Wesley, Reading, Mass., 1989.

[18] MATLAB, *High-Performance Numeric Computation and Visualization Software Reference Guide*, The MathWorks, Inc., Natick, Mass., 1992.

[19] S. Mori, C.Y. Suen, and K. Yamamoto, "Historical review of OCR research and development," *IEEE Proc.*, vol. 80, no. 7, pp. 1029-1058, 1992.

[20] M. Nakashima, T. Koezuka, N. Hiraoka, and T. Inagaki, "Automatic pattern recognition system with self-learning algorithm based on feature template matching," *Proc. SPIE*, vol. 635, pp. 480-486, 1986.

[21] H.K. Ramapriyan, "A multilevel approach to sequential detection of pictorial features," *IEEE Trans. Computers*, vol. 25, no. 1, pp. 66-78, Jan. 1976.

[22] S.V. Rice, J. Kanai, and T.A. Nartker, "A report on the accuracy of OCR devices," *Symp. Document Analysis and Information Retrieval*, Las Vegas, Nev., 1992.

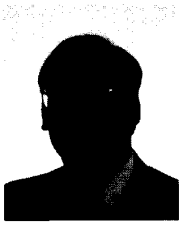
[23] A. Rosenfeld and A.C. Kak, *Digital Picture Processing*, Academic Press, New York, 1976.

[24] A. Rosenfeld and G.J. Vanderburg, "Coarse-fine template matching," *IEEE Trans. Systems, Man and Cybernetics*, vol. 7, pp. 104-107, 1977.

[25] M. Svedlow, C.D. McGillem, and P.E. Anuta, "Experimental examination of similarity measures and preprocessing methods used for image registration," *Symp. Machine Processing Remotely Sensed Data*, 1976.

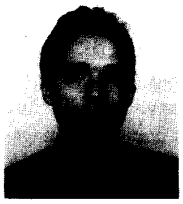
[26] J.T. Tou and R.C. Gonzalez, *Pattern Recognition Principles*. Addison-Wesley, Reading, Mass., 1974.

[27] G.J. Vanderburg and A. Rosenfeld, "Two-stage template matching," *IEEE Trans. Computers*, vol. 26, pp. 384-393, Apr. 1977.



Hadar I. Avi-Itzhak received the BSEE degree from the Technion-Israel Institute of Technology, Haifa, Israel, in 1984, and the MSEE and PhD from Stanford University, Stanford, Calif., in 1990 and 1994, respectively. He has been awarded the Hitachi Fellowship and the U.S. Air Force Fellowship.

Dr. Avi-Itzhak has been doing research, development, and consulting for industry and government in the fields of adaptive control, communication, and avionics, and has authored technical papers and patents on pattern recognition, neural networks, and image processing. He is currently a research scientist at Canon Research Center America, Palo Alto, Calif.



Jan A. Van Mieghem received his electrical engineering degree from the University KU Leuven, Belgium, and his MSEE degree from Stanford University where he was a BAEF Fellow. He is currently a PhD candidate in decision sciences at the Graduate School of Business at Stanford University, where he has been awarded the SIMA/Sloan Foundation Doctoral Fellowship.

Van Mieghem has published in the areas of pattern recognition, image processing, and applied probability.

Leonardo Rub received his BS in mathematics from the Massachussetts Institute of Technology, Cambridge, Mass., his MSCS from the Rensselaer Polytechnic Institute, Hartford, Conn., and an MS in scientific computing and computational mathematics from Stanford University.

Rub has been doing research, development, and consulting in the areas of statistics, operations research, digital signal processing, character recognition, and communications. He is currently with Octel Communications, Milpitas, Calif.