A Pattern Deformational Model and Bayes Error-Correcting Recognition System

WEN-HSIANG TSAI, STUDENT MEMBER, IEEE, AND KING-SUN FU, FELLOW, IEEE

Abstract—**A pattern deformational model is proposed in this paper. Pattern deformations are categorized into two types: local deformation and structural deformation. A structure-preserving local deformation can be decomposed into a syntactic deformation followed by a semantic deformation, the former being induced on primitive structures and the latter on primitive properties. Bayes error-correcting parsing algorithms are proposed accordingly which not only can perform normal syntax analysis but also can make statistical decisions. An optimum Bayes error-correcting recognition system is then formulated for pattern classification. The system can be considered as a hybrid pattern classifier which uses both syntactic and statistical pattern recognition techniques.**

I. Introduction

To RECOGNIZE noisy or deformed patterns using the syntactic approach, error-correcting parsing techniques using various decision criteria have been proposed [1]–[5], [20]. Errors induced on the primitives of noisy or deformed patterns usually are classified into three types: substitutions, deletions, and insertions. If only substitution errors are considered, the error-correcting parser is said to be structure-preserved. After an input pattern is parsed with respect to a certain pattern grammar, a quantitative measure, either deterministic or probabilistic, is computed by the parser to indicate the degree of possibility that the input pattern is generated by the grammar. The decision criterion is then used to assign the input pattern to the pattern class which corresponds to the minimum or maximum of the quantitative measure. Two most widely used decision criteria are minimum-distance and maximum-likelihood criteria, though others have also been proposed [2], [5].

Influenced by the linguistic type of representations which only adopt symbolic notations as terminals, most of the existing error-correcting parsing methods [1]–[4], [20] use discrete symbols to represent structural pattern primitives. However, it happens quite often that a primitive also contains continuous numerical information useful for pattern discrimination [5]–[7], [9]. For such cases these parsing methods are not powerful enough since they do not utilize continuous information. To take care of both structural and numerical information simultaneously, a deformational model for pattern primitives is introduced in this paper. Based on this model, error-correcting parsing and classification techniques using the Bayes decision rule are then proposed. Various known error-correcting parsing schemes and classification rules are compared with the proposed techniques. Illustrative examples are also given to show the practical feasibility of the proposed model and techniques.

Manuscript received September 20, 1978; revised June 28, 1979. This work was supported in part by NSF Grant ENG 78-16970 and ONR Contract 00014-79-C-0574.

The authors are with the School of Electrical Engineering, Purdue University, West Lafayette, IN 47907.

II. Basic Concepts

In this section we give a formal description of patterns, primitives, etc., used in syntactic pattern recognition from a broader point of view. Based on these concepts we propose a deformational model in the next section which will serve as a basis for developing a Bayes error-correcting recognition system.

An *observed pattern* usually can be considered as deformed from a *pure pattern* which is error free. For example, a smooth shape in a picture may become noisy after it is digitized. Here the original shape is the pure pattern and its noisy version is the observed pattern. When similar pure patterns are clustered as a pure pattern class, there corresponds a set of observed patterns. In practical applications, grammars are often inferred, either from pure or from observed patterns, to recognize observed patterns. In some simple cases, deformations, such as noises or distortions, existing in observed patterns can be eliminated by intensive preprocessing such as thresholding and smoothing. But in general they can not be eliminated entirely. This is why error-correcting parsing is necessary.

Before a class of patterns can be described by a pattern grammar, each pattern is decomposed into simpler structural units called primitives. Primitives should be chosen properly so that the resulting descriptions of the patterns using grammars can be simple [7]. We call the description of a pattern using some fixed primitives according to a certain preselected pattern structure as a *structural representation*, which is, for string languages, a *string* (*representation*) consisting of symbols each of which corresponds to a primitive, and is, for tree languages, a *tree* (*representation*) with each of its nodes corresponding to a primitive. Of course, pure primitives, pure patterns, and pure structural representations also have their corresponding observed primitives, observed patterns, and observed structural representations, respectively.

A detailed study of various kinds of primitives used for pattern descriptions [7]–[9] reveals that each primitive may contain two kinds of information, namely, the *syntactic information* and the *semantic information*. The syntactic

0018-9472/79/1200-0745$00.75 © 1979 IEEE

information gives a description of the primitive structure, and the semantic information provides logical or numerical descriptions of the primitive properties. For example, You and Fu [9] use two kinds of primitives—curve segment primitive and angle primitive—to describe shapes. The first is a curve segment with four numerical attributes to describe its direction, length, curvature, and symmetry. The second is an angle with one attribute to describe the angle amplitude. The resulting shape grammar is an attributed grammar. So, we consider a primitive a, either pure or observed, as a 2-tuple

$$a = (s, x) \tag{1}$$

where s is a *syntactic symbol* denoting the primitive structure of a, and $x = (x_1, x_2, \cdots, x_m)$ is an m-dimensional *semantic vector* with each x_i, $i = 1, 2, \cdots, m$, denoting a numerical measurement or a logical predicate, and $m \geq 0$. When $m = 0$, or no semantic information is available, set $x = \varnothing$ (empty vector). A similar idea was also proposed by Shaw [21] and described in Fu [7].

The primitives used in conventional syntactic pattern recognition tend to be restricted to symbolic representations which essentially provide only syntactic information. Even when a continuous type of numerical information, such as random noise, is included in the primitives, it is often thresholded into discrete levels which then are represented by a finite number of primitive symbols. Such an approach not only decreases the discrimination accuracy due to the numerical thresholding but also increases the number of grammar rules due to the increase of the number of primitives (i.e., terminals). With a primitive described as in (1), such a weakness could be eliminated. Also, since a primitive contains two kinds of information, we obtain a great deal of flexibility in selecting primitives [6]. Any structural unit can be selected as a primitive, and if more properties are needed to specify the primitive, numerical or logical attributes can be invoked. Furthermore, with semantic information separated from syntactic information in a primitive, a very systematic deformational model can be developed for optimum error-correcting parsing schemes which will be described in the following sections.

III. A Pattern Deformational Model

From previous discussions it is clear that a pattern or its structural representation ω can be characterized by a 2-tuple $\omega = (S, A)$ where $A = \{a_i \mid i = 1, 2, \cdots, n\}$ is a set of primitives used in ω and S denotes the pattern structure of ω together with implicitly assumed relations among the primitives. For discussion convenience, we assume that the subscripts for a_i are numbered according to some fixed order which is determined by the pattern structure S; when S is fixed, this ordering is also fixed.

A. Classification and Decomposition of Pattern Deformations

Given the structural representation $\omega = (S, A)$ of a certain pure pattern with pattern structure S and primitive set

$$A = \{a_i \mid a_i = (s_i, x_i),\ x_i = (x_{i1}, x_{i2}, \cdots, x_{iN_i}),\quad N_i \geq 0,\ i = 1, 2, \cdots, n\}, \tag{2}$$

the structural representation of one of its corresponding observed patterns, $\omega' = (S', A')$, with pattern structure S' and primitive set

$$A' = \{a'_i \mid a'_i = (s'_i, x'_i),\ x'_i = (x'_{i1}, x'_{i2}, \cdots, x'_{iN'_i}),\quad N'_i \geq 0,\ i = 1, 2, \cdots, n'\}, \tag{3}$$

can be considered as being transformed from ω through a series of deformations. Our deformational model categorizes all possible deformations into two major types: *structural deformation* and *local deformation.*

1) Local deformation—If $S = S'$ and so $n = n'$, but for some i, $i = 1, 2, \cdots, n$, $a_i \neq a'_i$, then we say ω' is *deformed locally* from ω. In other words, a local deformation induced on a pure pattern preserves the *entire* pattern structure but deforms some primitives *locally.* So a local deformation is also called a *structure-preserved deformation.* With respect to string representations, this simply means a *length-preserved deformation.*
2) Structural deformation—If $S \neq S'$, then we say that ω' is *deformed structurally* from ω. Various types of structural deformations, such as insertions, deletions, transpositions, and permutations [2], [11], [12], have been defined according to various kinds of structural difference between S and S'.

In this paper we deal only with local deformations, leaving structural deformations for further investigations. Let $a_i = (s_i, x_i)$ be a deformed primitive where

$$x_i = (x_{i1}, x_{i2}, \cdots, x_{iN_i}), \tag{4}$$

and $c_i = (t_i, z_i)$ be one of its observed versions, where

$$z_i = (z_{i1}, z_{i2}, \cdots, z_{iN_i}). \tag{5}$$

At least two types of local deformation can be identified as follows.

1) Syntactic local deformation (syn.l.d.)—This is the case when $t_i \neq s_i$. In other words, when the primitive structure is changed to another one, a *syntactic local deformation* is induced, which usually is called a *substitution error.*
2) Semantic local deformation (sem.l.d.)—When the local deformation on a_i does not change the primitive structure but only corrupts the semantic information, i.e., when $t_i = s_i$ but $z_i \neq x_i$, then it is called a *semantic local deformation.* If every primitive used by a pattern has an identical primitive structure, then every local deformation is semantic.

In general, we can consider a local deformation as a two-step transformation from $a_i = (s_i, x_i)$ to $c_i = (t_i, z_i)$ described by (6).

$$\underset{\text{pure primit.}}{a_i = (s_i, x_i)} \xrightarrow[\text{syn.l.d.}]{} \underset{\text{semi-pure primit.}}{b_i = (t_i, y_i)} \xrightarrow[\text{sem.l.d.}]{} \underset{\text{observed primit.}}{c_i = (t_i, z_i)}, \tag{6}$$

where $b_i = (t_i, y_i)$, called a *semi-pure primitive*, is created to denote one of the syntactically local-deformed versions of (s_i, x_i) with y_i being a *representative semantic vector* for t_i, which is created only for explanatory convenience and does not have much practical use later in our derivation of parsing procedures.[1] When $t_i = s_i$, then $y_i = x_i$ and only a semantic local deformation occurs in the two-step transformation.

B. Pattern Deformation Probability or Density Function

Let $A = \{a_i \mid a_i = (s_i, x_i), i = 1, 2, \cdots, n\}$ denote all the pure primitives used in a pure pattern. Though each a_i can be deformed *syntactically* into a set of semi-pure primitives $D_{a_i} = \{b_{ij} \mid b_{ij} = (t_{ij}, y_{ij}), j = 1, 2, \cdots, k_i\}$, each deformation $a_i \rightarrow b_{ij}$ may occur with a different probability. So there exists a conditional probability function p defined on D_{a_i} such that $p(b_{ij} \mid a_i) = p(t_{ij} \mid s_i)$ is the probability for s_i to be deformed into $t_{ij}, j = 1, 2, \cdots, k_i$. Similarly, since each b_{ij} can be deformed *semantically* into a set of observed primitives $D_{b_{ij}} = \{c_{ijk} \mid c_{ijk} = (t_{ij}, z_{ijk}), z_{ijk} \in R_{ij}\}$, where R_{ij} is a finite or infinitive range for z_{ijk}, we can define a conditional probability or density function q on $D_{b_{ij}}$ such that $q(z_{ijk} \mid b_{ij}, a_i) = q(z_{ijk} \mid t_{ij}, s_i)$ is the probability or density function for $b_{ij} = (t_{ij}, y_{ij})$ to be deformed into $c_{ijk} = (t_{ij}, z_{ijk})$. Therefore, from the statistical point of view, a local deformation of $a_i = (s_i, x_i)$ into $c_{ijk} = (t_{ij}, z_{ijk})$ now can be interpreted as the following:

$$a_i = (s_i, x_i) \xrightarrow[\text{syn.l.d.}]{p(t_{ij}|s_i)} b_{ij} = (t_{ij}, y_{ij}) \xrightarrow[\text{sem.l.d.}]{q(z_{ijk}|t_{ij}, s_i)} c_{ijk} = (t_{ij}, z_{ijk}) \tag{7}$$

where $p(\cdot \mid s_i)$ is the conditional probability function given a_i (or s_i) defined on D_{a_i}, and $q(\cdot \mid t_{ij}, s_i)$ is the conditional probability or density function given a_i and b_{ij} (or s_i, t_{ij}) defined on $D_{b_{ij}}$. We also assume that $a_i \in D_{a_i}$, and $b_{ij} \in D_{b_{ij}}$.

To be more specific, we give two examples with semantic local deformation, assuming that no syntactic local deformation is involved—that is,

$$a_i = (s_i, x_i) \xrightarrow[\text{sem.l.d.}]{q(z_{ij}|s_i)} c_{ij} = (s_i, z_{ij}). \tag{8}$$

1) Continuous random noise—This is the case when the semantic vector x_i in a pure primitive $a_i = (s_i, x_i)$ is corrupted by random noise. The deformed or noisy version of x_i, denoted as z_{ij} above, is actually a vector-valued random variable z_{ij} with continuous density function $q(z_{ij} \mid s_i)$. If the noise associated with z_{ij} is normally distributed with zero mean, then x_i in fact is just the mean vector of z_{ij}, or $x_i = E\{z_{ij}\}$.
2) Discrete distortion variations—In some cases, x_i may be deformed into only a finite number of observed versions z_{ij}. Then $q(z_{ij} \mid s_i)$ above is just a discrete probability function defined on all possible z_{ij}.

[1] Sometimes for normally distributed z_i, y_i can be conveniently chosen to be the mean value of z_i.

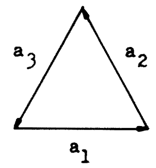

Fig. 1. A pure pattern ω—a unit equilateral triangle.

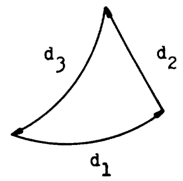

Fig. 2. An observed pattern ω'—a deformed triangle.

Back to our discussion of two-step local deformations, given a pure primitive $a_i = (s_i, x_i)$, the probability or density function that it is deformed locally into an observed primitive $c_i = (t_i, z_i)$ now can be computed as

$$r(c_i \mid a_i) = p(t_i \mid s_i) q(z_i \mid t_i, s_i). \tag{9}$$

We will call $r(c_i \mid a_i)$ the *primitive deformation probability or density function* of c_i from a_i. For a pure pattern $\omega = (S, A)$ with $A = \{a_i \mid a_i = (s_i, x_i), i = 1, 2, \cdots, n\}$, the probability or density function that ω is deformed locally into a structure-preserved observed pattern $\omega' = (S, C)$ with $C = \{c_i \mid c_i = (t_i, z_i), a_i \xrightarrow[\text{l.d.}]{} c_i, i = 1, 2, \cdots, n\}$ is then

$$p(\omega' \mid \omega) = \prod_{i=1}^{n} r(c_i \mid a_i) \tag{10}$$

or

$$p(\omega' \mid \omega) = \prod_{i=1}^{n} p(t_i \mid s_i) q(z_i \mid t_i, s_i), \tag{11}$$

if each a_i is deformed *independently* into c_i, $i = 1, 2, \cdots, n$. Such independence assumption for local deformations of primitives was also considered by Grenander [13], Kovalevsky [14], and Fung and Fu [3]. $p(\omega' \mid \omega)$ is called the *pattern deformation probability or density function* of ω' from ω. An example is given in the following to illustrate the previous discussions and clarify the notations used.

Example 1) Deformation of an equilateral triangle: Suppose that the pure pattern we are dealing with is a unit equilateral triangle as shown in Fig. 1. The primitives we choose naturally are the edges—line segments. Now, due to local deformations, each line segment may be deformed syntactically into two kinds of curve segments—one kind with a fixed positive curvature and the other with a fixed negative curvature. Furthermore, each line or curve segment may be deformed semantically on its length[2] and direction (with respect to the x-axis) by normally distributed random noise with zero mean. So the pure pattern, an equilateral triangle, is subject to size and orientation variations. A possible observed pattern might be like the one shown in Fig. 2.

[2] Here the length of a curve segment is defined to be the length of the line segment joining the two end points of the curve.

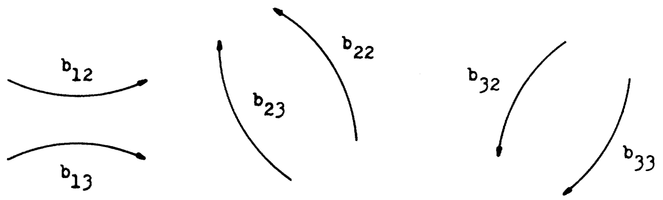

Fig. 3. Semi-pure primitives for Fig. 1.

More specifically, using L, C_p, and C_n as syntactic symbols to specify the three primitive structures—line segments, curve segments with positive curvatures, and curve segments with negative curvatures, respectively, we have the following three pure primitives for the edges in the form of 2-tuples:

$$a_1 = (L, x_1),\ x_1 = (1, 0°),$$
$$a_2 = (L, x_2),\ x_2 = (1, 120°),$$
$$a_3 = (L, x_3),\ x_3 = (1, 240°),$$

where $x_i = (x_{i1}, x_{i2})$ with $x_{i1} = 1$ (unit edge length) and $x_{i2} = (i-1) \cdot 120°$ (edge direction) is the semantic vector of a_i, $i = 1, 2, 3$, and the following six semi-pure primitives (shown in Fig. 3):

$$b_{12} = (C_p, y_{12}),\ y_{12} = (1, 0°),$$
$$b_{13} = (C_n, y_{13}),\ y_{13} = (1, 0°),$$
$$b_{22} = (C_p, y_{22}),\ y_{22} = (1, 120°),$$
$$b_{23} = (C_n, y_{23}),\ y_{23} = (1, 120°),$$
$$b_{32} = (C_p, y_{32}),\ y_{32} = (1, 240°),$$
$$b_{33} = (C_n, y_{33}),\ y_{33} = (1, 240°),$$

where y_{ij} is the representative semantic vector of b_{ij}, $i = 1, 2, 3$, $j = 2, 3$. Since a_i is considered as a deformed version of itself, we have

$$D_{a_i} = \{b_{i1} = a_i, b_{i2}, b_{i3}\},$$

with the following assigned probabilities

$$p(b_{i1} | a_i) = p(L | L) = 0.7,$$
$$p(b_{i2} | a_i) = p(C_p | L) = 0.2,$$
$$p(b_{i3} | a_i) = p(C_n | L) = 0.1, \qquad \text{for } i = 1, 2, 3.$$

Furthermore, we have

$$D_{b_{ij}} = \{c_{ijk} | c_{ijk} = (T_j, z_{ijk}),\ z_{ijk} = (l_{ijk}, \theta_{ijk})\},$$

with

$$T_j = L, \qquad j = 1,$$
$$= C_p, \qquad j = 2,$$
$$= C_n, \qquad j = 3,$$

and the following assigned density functions:

$$q(l_{ijk} | b_{ij}, a_i) = \frac{1}{\sqrt{2\pi}\,\sigma_{1j}} \exp\left[-\tfrac{1}{2}(l_{ijk} - l_0)^2/\sigma_{1j}^2\right],$$

$$q(\theta_{ijk} | b_{ij}, a_i) = \frac{1}{\sqrt{2\pi}\,\sigma_{2j}} \exp\left[-\tfrac{1}{2}(\theta_{ijk} - \theta_{0i})^2/\sigma_{2j}^2\right],$$

with $l_0 = 1$, $\theta_{0i} = (i-1) \cdot 120°$ for $i = 1, 2, 3$,

$$\sigma_{1j} = 0.1,\ \sigma_{2j} = 4° \qquad \text{for } j = 1,$$

and

$$\sigma_{1j} = 0.2,\ \sigma_{2j} = 6° \qquad \text{for } j = 2, 3.$$

l_{ijk} and θ_{ijk} are assumed to be independently distributed, i.e.,

$$q(z_{ijk} | b_{ij}, a_i) = q(l_{ijk} | b_{ij}, a_i) \cdot q(\theta_{ijk} | b_{ij}, a_i).$$

Now, we want to compute the pattern deformation density function of ω' from ω. ω and ω' can be specified as 2-tuples, $\omega = (S, A)$ with pattern structure S and primitive set $A = \{a_1, a_2, a_3\}$, and $\omega' = (S, B)$ with the same pattern structure S and primitive set $B = \{d_1, d_2, d_3\}$, where

$$d_1 = (C_p, w_1),\ w_1 = (1.1, 10°),$$
$$d_2 = (L, w_2),\ w_2 = (0.9, 105°),$$
$$d_3 = (C_n, w_3),\ w_3 = (1.2, 235°).$$

The result is

$$\begin{aligned}
p(\omega' | \omega) &= \prod_{i=1}^{3} r(d_i | a_i) \\
&= p(C_p | L) q(w_1 | C_p, L) \cdot p(L | L) q(w_2 | L, L) \\
&\quad \cdot p(C_n | L) q(w_3 | C_n, L) \\
&= p(b_{12} | a_1) q(w_1 | b_{12}, a_1) \\
&\quad \cdot p(b_{21} | a_2) q(w_2 | b_{21}, a_2) \\
&\quad \cdot p(b_{33} | a_3) q(w_3 | b_{33}, a_3) \\
&= 0.2 \cdot \frac{1}{\sqrt{2\pi} \cdot 0.2} \exp\left[-\tfrac{1}{2}(1.1 - 1.0)^2/0.2^2\right] \\
&\quad \cdot \frac{1}{\sqrt{2\pi} \cdot 6} \exp\left[-\tfrac{1}{2}(10 - 0)^2/6^2\right] \\
&\quad \cdot 0.7 \cdot \frac{1}{\sqrt{2\pi} \cdot 0.1} \exp\left[-\tfrac{1}{2}(0.9 - 1.0)^2/0.1^2\right] \\
&\quad \cdot \frac{1}{\sqrt{2\pi} \cdot 4} \exp\left[-\tfrac{1}{2}(105 - 120)^2/4^2\right] \\
&\quad \cdot 0.1 \cdot \frac{1}{\sqrt{2\pi} \cdot 0.2} \exp\left[-\tfrac{1}{2}(1.2 - 1.0)^2/0.2^2\right] \\
&\quad \cdot \frac{1}{\sqrt{2\pi} \cdot 6} \exp\left[-\tfrac{1}{2}(235 - 240)^2/6^2\right] \\
&= 4.95 \times 10^{-9}.
\end{aligned}$$

IV. Bayes Structure-Preserved Error-Correcting Parsers

In this section we derive structure-preserved error-correcting parsers (SPECP) optimum in the Bayes sense for locally deformed patterns. Given a pattern class consisting of various pure patterns which can be generated by a pattern grammar, we can, from the statistical point of view, consider each pure pattern together with all its possible locally deformed versions as a distinct *subclass* of the given pattern class. Then the SPECP's to be derived, which we will call Bayes SPECP's, are optimum in the sense that, in addition to possessing syntactic parsing capability, they are *Bayes subclass classifiers* which assign each given observed pattern to a subclass according to the Bayes decision rule.

A. Bayes Decision Rule and Bayes Distances

Given an observed pattern $\omega = (S, A)$ with $A = \{a_i \mid a_i = (s_i, x_i), x_i = (x_{i1}, x_{i2}, \cdots, x_{iL_i}), i = 1, 2, \cdots, n\}$ of a certain pure pattern class C which consists, for simplicity, of only two pure patterns $\omega_1 = (S, B_1)$ and $\omega_2 = (S, B_2)$ with $B_1 = \{b_i^1 \mid b_i^1 = (t_i^1, y_i^1), y_i^1 = (y_{i1}^1, y_{i2}^1, \cdots, y_{iM_{i,1}}^1), i = 1, 2, \cdots, n\}$ and $B_2 = \{b_i^2 \mid b_i^2 = (t_i^2, y_i^2), y_i^2 = (y_{i1}^2, y_{i2}^2, \cdots, y_{iM_{i,2}}^2), i = 1, 2, \cdots, n\}$, we want to assign ω to one of the two pure pattern subclasses ω_1 and ω_2 according to the theory of statistical hypothesis testing. Using the Bayes decision rule, we get, according to the analysis for the deformational model in Section III under the independence assumption for local deformations,

$$\text{Decide} \quad \omega \sim \begin{cases} \omega_1 \\ \omega_2 \end{cases}, \quad \text{if } \frac{P(\omega_1 \mid \omega)}{P(\omega_2 \mid \omega)} \gtrless 1$$

or

$$\begin{aligned} \text{Decide } \omega \sim \begin{cases} \omega_1 \\ \omega_2 \end{cases}, \quad & \text{if } \frac{p(\omega \mid \omega_1) P(\omega_1)}{p(\omega \mid \omega_2) P(\omega_2)} \\ &= \left[\prod_{i=1}^{n} \frac{r(a_i \mid b_i^1)}{r(a_i \mid b_i^2)} \right] \cdot \frac{P(\omega_1)}{P(\omega_2)} \\ &= \left[\prod_{i=1}^{n} \frac{p(s_i \mid t_i^1) q(x_i \mid s_i, t_i^1)}{p(s_i \mid t_i^2) q(x_i \mid s_i, t_i^2)} \right] \cdot \frac{P(\omega_1)}{P(\omega_2)} \gtrless 1. \end{aligned} \tag{12}$$

After taking logarithms, we obtain

$$\begin{aligned} &\text{Decide} \quad \omega \sim \begin{cases} \omega_1 \\ \omega_2 \end{cases}, \\ &\text{if } \sum_{i=1}^{n} [\ln p(s_i \mid t_i^1) + \ln q(x_i \mid s_i, t_i^1)] + \ln P(\omega_1) \\ &\gtrless \sum_{i=1}^{n} [\ln p(s_i \mid t_i^2) + \ln q(x_i \mid s_i, t_i^2)] + \ln P(\omega_2) \end{aligned} \tag{13}$$

where $P(\omega_1 \mid \omega)$, $P(\omega_1 \mid \omega)$, $P(\omega_1)$, $P(\omega_2)$ are *a posteriori* and *a priori* probabilities for pure pattern subclass ω_1 and ω_2. When the pure pattern class C consists of more than two patterns, the above decision rule can be extended as follows. Let λ_j be such that

$$-\ln \lambda_j = -\sum_{i=1}^{n} [\ln p(s_i \mid t_i^j) + \ln q(x_i \mid s_i, t_i^j)] - \ln P(\omega_j), \tag{14}$$

$j = 1, 2, \cdots, M$, with M, either finite or infinite, being the total number of pure patterns in C. Then decide $\omega \sim \omega_k$ if k is such that

$$-\ln \lambda_k = \min_{j=1,2,\cdots,M} (-\ln \lambda_j). \tag{15}$$

We call the term $-\ln \lambda_j$ the *Bayes distance* $B(\omega, \omega_j)$ from ω to ω_j, and the term $-\ln \lambda_k$ the *minimum Bayes distance* $B(\omega, C)$ from ω to the pure pattern class C.

With the Bayes distance defined, the *Bayes SPECP*, constructed from the pattern grammar G_c for the given pure pattern class C, is used to search, for the given input observed pattern ω, a pure pattern ω_k accepted by G_c with a minimum Bayes distance $B(\omega, \omega_k) = B(\omega, C)$. So our problem now is reduced to the computation of Bayes distances $-\ln \lambda_j$ during parsing. Since the parsing will pass each primitive at least once, there is no problem in computing the first term $\sum_{i=1}^{n} [p(s_i \mid t_i^j) + \ln q(x_i \mid s_i, t_i^j)]$ in (14), as will be seen later. But how to get the information about the *a priori* probability $P(\omega_j)$ for the pure pattern ω_j during parsing is on the contrary not so obvious. The solution is to use a *stochastic grammar* for the pattern class C.

B. Use of Stochastic Grammars for Computing Pattern Probabilities

Stochastic grammars have been introduced to characterize noisy patterns including the probability of occurrence for each pattern generated by the pattern grammars [7]. This property is exactly what we want for computing pattern probabilities $P(\omega_j)$. To be more specific, a stochastic grammar is a grammar each of whose production rules is associated with an occurrence probability. When a stochastic pattern grammar is used to generate the structural representation of a given pattern, a pattern occurrence probability is also generated simultaneously, which is the product of all the production rule probabilities used in deriving the structural representation. For details, see Fu [7]. And for inference of production rule probabilities, see Lee and Fu [15]. Here we briefly review the basic notations and definitions of stochastic context-free string grammars and stochastic tree grammars [7], [17].

Definition 1: A stochastic context-free string grammar is a 4-tuples $G_s = (V_N, V_T, P_s, S)$, where V_N is a finite set of nonterminals, V_T is a finite set of terminals, S is a start symbol, P_s is a finite set of stochastic production rules, each of which is of the form

$$A_i \xrightarrow{p_{ij}} \alpha_{ij}, j = 1, 2, \cdots, n_i, i = 1, 2, \cdots, l, \tag{16}$$

where $A_i \in V_N$, $\alpha_{ij} \in (V_T \cup V_N)^*$, n_i is the number of distinct production rules with A_i at the left side, l is the number of nonterminals, and p_{ij} is the probability associated with this production rule. Furthermore,

$$0 < p_{ij} \le 1 \quad \text{and} \quad \sum_{j=1}^{n_i} p_{ij} = 1. \tag{17}$$

Definition 2: A stochastic context-free string grammar G_s is in Chomsky normal form if each of its production rules is of the form

$$A \xrightarrow{p} BC \text{ or } A \xrightarrow{p} a$$

where $A, B, C \in V_N$, $a \in V_T$.

Definition 3: A stochastic tree grammar over $\langle V_T, r \rangle$ in its expansive form is a 4-tuple $G_t = (V_N \cup V_T, r, P, S)$, where V_N,

V_T, S are the same as defined in Definition 1, $r: V_T \to N$, the set of nonnegative integers, is a rank function denoting the number of direct descendants of a node with a symbol in V_T as its label, and P is a set of stochastic production rules, each of which is in the form

$$X_i \xrightarrow{p_{ij}} \begin{array}{c} a_j \\ \diagup \; | \; \diagdown \\ X_{ij1}\, X_{ij2} \cdots X_{ijr(a_j)} \end{array} \quad \text{or} \quad X_i \xrightarrow{p_{ij}} a_j \quad (18)$$

where $a_j \in V_T$, X_i, X_{ij1}, X_{ij2}, $\cdots$, $X_{ijr(a_j)} \in V_N$, $1 \le j \le n_i$, $1 \le i \le l$, n_i, l, p_{ij} are the same as defined in Definition 1, and

$$0 < p_{ij} \le 1 \qquad \text{and} \qquad \sum_{j=1}^{n_i} p_{ij} = 1.$$

C. Bayes SPECP for String Languages

We describe in the following a Bayes SPECP for context-free string languages. Given a stochastic context-free string grammar $G_s = (V_N, V_T, P_s, S)$ for a pure pattern class, assume that the terminal set $V_T = \{a_i | a_i = (t_i, w_i),\ i = 1, 2, \cdots, l\}$ contains all possible pure primitives used by the pure patterns. For each a_i, $i = 1, 2, \cdots, l$, let $p(\cdot | a_i) = p(\cdot | t_i)$ be the conditional probability function defined on $D_{a_i} = \{b_{ij} | b_{ij} = (u_{ij}, y_{ij}),\ a_i \xrightarrow[\text{syn.l.d.}]{} b_{ij},\ j = 1, 2, \cdots, k_i\}$, and $q(\cdot | a_i, b_{ij}) = q(\cdot | t_i, u_{ij})$ be the conditional probability or density function defined on $D_{b_{ij}} = \{c_{ijk} | c_{ijk} = (u_{ij}, z_{ijk}),\ b_{ij} \xrightarrow[\text{sem.l.d.}]{} c_{ijk},\ z_{ijk} \in R_{ij}\}$. Let

$$V'_T = \bigcup_{i=1}^{l} \left[\bigcup_{j=1}^{k_i} D_{b_{ij}} \right] \quad (19)$$

denote all possible deformed primitives, and note that $V_T \subset V'_T$. The following algorithm for the Bayes SPECP is a modified Cocke–Yonger–Kasami parsing algorithm [16], which essentially tries to construct a parse table T for an input observed string representation y and then parses through the table to obtain a pure string representation x with a minimum Bayes distance $B(y, x)$. The table T consists of entries t_{ij}, $1 \le i \le n$, $1 \le j \le n - i + 1$, where n is the length of string y. Each t_{ij} is a set of triplets (A, d, k), where $A \in V_N$ is an intermediate nonterminal used in deriving x, $d \in (0, \infty)$ is part of the Bayes distance, and k specifies the product rule used with A at its left side.

Algorithm 1: Bayes structure-preserved error-correcting parser for string languages.

Input: A stochastic context-free string grammar $G_s = (V_N, V_T, P_s, S)$ in Chomsky normal form without ε-productions, and an observed string representation $y \in V_T'^{*}$, $y = c_1 c_2 \cdots c_n$, $c_i = (s_i, x_i)$, $i = 1, 2, \cdots, n$.

Output: A pure string representation x accepted by G_s with a minimum Bayes distance $B(y, x)$, if y is structure-preserved.

Method: Label all production rules and let $k: A \xrightarrow{p} \alpha$ denote that $A \xrightarrow{p} \alpha$ is the kth rule in P_s.

Step 1: Construct t_{i1} for each i, $i = 1, 2, \cdots, n$. Let $A \in V_N$. For every $k_h: A \xrightarrow{p_h} a_h$ in P_s, $h = 1, 2, \cdots, n_A$, where $a_h = (t_h, w_h)$, n_A is the number of production rules each with A on the left side and a terminal on the right side, let

$$d_{ih} = -[\ln p(s_i | t_h) + \ln q(x_i | t_h, s_i) + \ln p_h], \quad (20)$$

$i = 1, 2, \cdots, n$. Then set

$$t_{i1} = \{(A, d_{il}, k_l) | d_{il} = \min_{h = 1, 2, \cdots, n_A} d_{ih},\ A \in V_N\}. \quad (21)$$

Step 2: Construct t_{ij}, $j = 2, \cdots, n$, inductively. Assume that $t_{ij'}$ has been computed for all i, $1 \le i \le n$, and for all j', $1 \le j' < j$. For every $k_h: A \xrightarrow{p_h} B_h C_h$, $h = 1, 2, \cdots, n'_A$, where n'_A is the number of production rules with A on the left side and two nonterminals on the right side, if there exists some m, $1 \le m < j$, such that $(B_h, e_{h1}, k_{h1}) \in t_{im}$ and $(C_h, e_{h2}, k_{h2}) \in t_{i+m, j-m}$, let $e_{ih} = e_{h1} + e_{h2} - \ln p_h$. Then set

$$t_{ij} = \{(A, e_{il}, k_l) | e_{il} = \min_{h = 1, 2, \cdots, n_A'} e_{ih},\ A \in V_N\}. \quad (22)$$

Step 3: Repeat Step 2 until t_{ij} is computed for all $1 \le i \le n$ and $1 \le j \le n - i + 1$.

Step 4: When the entire table T is completed, examine entry t_{1n}. If there exists a triplet (S, d, k) in t_{1n} for some d and k, then set $B(y, x) = d$, and the desired pure string representation x can be easily traced out from the parse table T, starting from the kth production rule. If no (S, d, k) exists in t_{1n}, then input observed string representation y is not structure-preserved.

D. Bayes SPECP for Tree Languages

Using the minimum-Bayes-distance criterion again, we propose a Bayes SPECP for tree languages. Given a stochastic tree grammar $G_s = (V_N \cup V_T, r, P_s, S)$ over $\langle V_T, r \rangle$ in its expansive form, let V_T, $p(\cdot | a_i) = p(\cdot | t_i)$, $q(\cdot | a_i, b_{ij}) = q(\cdot | t_i, u_{ij})$, D_{a_i}, $D_{b_{ij}}$, and V'_T be all the same as those defined in Section IV-C. The following algorithm for the Bayes SPECP follows the concept of tree automata [17], and is a backward procedure for constructing a tree-like transition table T for an input observed tree representation β. Let the tree structure (i.e., the tree domain) of β be denoted as D_β, then corresponding to each node b in D_β is an entry t_b in T, which consists of a set of triplets (A, d, k), where $A \in V_N$ is a candidate state for node b, d is part of the Bayes distance, and k specifies the production rule used with A at its left side.

Algorithm 2: Bayes structure-preserved error-correcting parser for tree languages.

Input: A stochastic tree grammar $G_s = (V_N \cup V_T, r, P_s, S)$ over $\langle V_T, r \rangle$ in its expansive form, and an observed tree representation β with $\beta(b) = (S_b, x_b)$ as its observed primitive at node b, $(s_b, x_b) \in V'_T$.

Output: A pure tree representation α accepted by G_s with a minimum Bayes distance $B(\beta, \alpha)$, if β is structure-preserved.

Method: Let $t_{b \cdot i}$ denote the entry in T, which consists of the set of triplets corresponding to the ith descendant of node b.

Step 1: For each node b in β such that $r[\beta(b)] = 0$ (i.e., b has no descendant), add to t_b a triplet (A, d, k) with

$$d = -[\ln p(s_b | t_k) + \ln q(x_b | t_k, s_b) + \ln p_k], \quad \text{if } A \xrightarrow{p_k} a_k \quad (23)$$

with $a_k = (t_k, w_k)$ is the kth production rule in P_s.

Step 2: For each node b in β such that $r[\beta(b)] = N \neq 0$, add to t_b a triplet (A, d_0, k) with

$$d_0 = -[\ln p(s_b | t_k) + \ln q(x_b | t_k, s_b) + \ln p_k] + d_1 + d_2 + \cdots + d_N, \quad \text{if } A \xrightarrow{p_k} \begin{array}{c} a_k \\ / \ \backslash \\ A_1 \cdots A_N \end{array} \quad (24)$$

with $a_k = (t_k, w_k)$ is the kth production rule in P_s and $(A_1, d_1, k_1) \in t_{b \cdot 1}$, $(A_2, d_2, k_2) \in t_{b \cdot 2}, \cdots, (A_N, d_N, k_N) \in t_{b \cdot N}$.

Step 3: For any two triplet (B_i, d_i, k_i), (B_j, d_j, k_j) in each t_b, delete the former if $d_i \geq d_j$, or the latter if $d_i < d_j$.

Step 4: Repeat Steps 1–3 until all nodes in β have been processed.

Step 5: Examine t_0, the root entry of the transition table T. If $(S, d, k) \in t_0$ for some d and k, then set $B(\beta, \alpha) = d$, and the desired pure tree representation α can be easily traced out from T, starting from the kth production rule in P_s. If no (S, d, k) exists in t_0, then the input observed tree representation β is not structure-preserved.

E. Comments on Various SPECP and Least-Square-Error Distance Criteria

Fung and Fu [3] have proposed a maximum-likelihood SPECP for string languages, but the grammars used are nonstochastic, so their SPECP is suboptimal with the assumption that all pattern subclasses occur with equal probability. SPECP's using stochastic grammars has been proposed by Fung and Fu [18], Lu and Fu [20], and Thompson [2], but from the view point of our deformational model, these SPECP's consider only syntactic local deformations, and so are limited in their usage in syntactic pattern recognition problems where useful semantic information, especially when it is continuous, is contained in the pattern primitives.[3] Of course, these SPECP's still can be used to handle continuous types of semantic information by thresholding them into finite discrete levels, but obviously this will decrease the error-correcting recognition accuracy of the SPECP's, as mentioned previously in Section II, and as will be shown by an example in Section V-D.

Next, SPECP's for string and tree languages using the *minimum-distance* criterion have also been proposed [1], [4]. In addition to being limited to syntactic local deformations, these SPECP's are statistically optimum only under very special conditions, although they are convenient and important in practical applications where deformation probability or density functions are difficult to infer.

Finally, we propose in the following a new criterion, namely, the *least-square-error* (*LSE*) *distance* criterion for SPECP's, which is a special case of the minimum-Bayes-distance criterion but is useful for semantic local deformations where the observed semantic vector in a primitive is normally distributed. Such cases often occur when patterns are corrupted with random noise.

Assuming that no syntactic local deformation is involved, we want to derive the Bayes distance between a pure pattern $\omega = (S, B)$ and one of its locally deformed observed patterns, $\omega' = (S, A)$, where $A = \{a_i | a_i = (s_i, x_i), x_i = (x_{i1}, x_{i2}, \cdots, x_{iN}), i = 1, 2, \cdots, n\}$ and $B = \{b_i | b_i = (s_i, w_i), w_i = (w_{i1}, w_{i2}, \cdots, w_{iN}), i = 1, 2, \cdots, n\}$, under the following conditions.

1) Component random variables x_{ij} of x_i are all independently and normally distributed with mean w_{ij} and variance σ_{ij}^2, $j = 1, 2, \cdots, N$:

$$f_{ij}(x_{ij}) = \frac{1}{\sqrt{2\pi}\,\sigma_{ij}} \exp\left[-\tfrac{1}{2}(x_{ij} - w_{ij})^2/\sigma_{ij}^2\right]. \quad (25)$$

An example for this case happens when every x_i is corrupted by random noise with zero mean and variance σ_{ij}^2.

2) The pure pattern ω occurs with the same probability as any other, so that $P(\omega)$ is a constant for every pure pattern ω.

Then the Bayes distance from ω' to ω is

$$\begin{aligned} B(\omega', \omega) &= -\ln \lambda \\ &= -\sum_{i=1}^{n} [\ln p(s_i | s_i) + \ln q(x_i | s_i, s_i)] - \ln P(\omega) \\ &= -\sum_{i=1}^{n} \left(\sum_{j=1}^{N} \ln f_{ij}(x_{ij}) \right) - \ln P(\omega) \\ &= K + \sum_{i=1}^{n} \sum_{j=1}^{N} \cdot \left[\frac{1}{2} \left(\frac{x_{ij} - w_{ij}}{\sigma_{ij}} \right)^2 + \ln \sigma_{ij} \right], \end{aligned} \quad (26)$$

where K is a constant, and as far as discrimination is concerned, we can define the *normalized square-error distance* as

$$B_1(\omega', \omega) = \sum_{i=1}^{n} \sum_{j=1}^{N} \left[\left(\frac{x_{ij} - w_{ij}}{\sigma_{ij}} \right)^2 + 2 \ln \sigma_{ij} \right], \quad (27)$$

and the (*unnormalized*) *square-error distance* as

$$B_2(\omega', \omega) = \sum_{i=1}^{n} \sum_{j=1}^{N} (x_{ij} - w_{ij})^2 \quad (28)$$

which is appropriate under a further assumption that all $\sigma_{ij} = 1$. A SPECP using the normalized or unnormalized least-square-error (LSE) distance criterion is called a *normalized* or *unnormalized LSE SPECP*. These two kinds of LSE SPECP's for tree languages have been used by Tsai and Fu [5] for picture segmentation and discrimination of textures corrupted by random noise, and the result obtained from the normalized LSE SPECP, as expected, is better than that from the unnormalized version.

[3] Deletion and insertion errors are also considered in [2], [20].

V. Bayes Error-Correcting Recognition System—A Hybrid Pattern Classifier

What we have investigated so far is just one-class classification problems: given an unknown pattern ω' and a pattern grammar G, we can use the Bayes SPECP of G to parse ω' and find out a pure pattern ω which is accepted by G and is the closest to ω' in the sense of Bayes distances. In this section we will study multi-class problems, i.e., given m pattern classes $C_1, C_2, \cdots, C_m$ of pure patterns and their pattern grammars $G_1, G_2, \cdots, G_m$, we want to assign ω', a given observed pattern, to one of these m classes according to the Bayes decision rule.

A. A Multi-Class Bayes Recognition System

Applying the Bayes decision rule, we get

$$\text{Decide } \omega' \sim C_l \text{ if } P(C_l|\omega') = \max_{i=1,2,\cdots,m} P(C_i|\omega'), \tag{29}$$

or

$$\text{Decide } \omega' \sim C_l \text{ if}$$
$$p(\omega'|C_l)P(C_l) = \max_{i=1,2,\cdots,m} p(\omega'|C_i)P(C_i) \tag{30}$$

where $P(C_i)$ is the *a priori* probability of class C_i, and $p(\omega'|C_i)$ is the conditional probability or density function of ω' given that $\omega' \in C_i$. Due to the possible ambiguity existing in G_i and the error-correcting capability of the parsing algorithm, ω' may be derived by G_i in several different ways, or, in other words, ω' may be regarded as deformed from several pure patterns $\omega_1, \omega_2, \cdots, \omega_{k_i} \in C_i$. So

$$p(\omega'|C_i) = \sum_{\substack{j=1 \\ \omega_j \in C_i}}^{k_i} p(\omega'|\omega_j)P(\omega_j|C_i) \tag{31}$$

and the multi-class classification rule now becomes

$$\text{Decide } \omega' \sim C_l \text{ if}$$
$$\left[\sum_{\substack{j=1 \\ \omega_j \in C_l}}^{k_l} p(\omega'|\omega_j)P(\omega_j|C_l)\right]P(C_l) = \max_{i=1,2,\cdots,m}\left[\sum_{\substack{j=1 \\ \omega_j \in C_i}}^{k_i} p(\omega'|\omega_j)P(\omega_j|C_i)\right]P(C_i). \tag{32}$$

where $P(\omega_j|C_i)$ can be obtained from stochastic pattern grammars (denoted as $P(\omega_j)$ in Section IV for one-class problems). Note that in (32), each $p(\omega'|\omega_j)P(\omega_j|C_i)$ is related to the Bayes distance $B(\omega', \omega_j)$ by the following equality:

$$p(\omega'|\omega_j)P(\omega_j|C_i) = \exp\left[-B(\omega', \omega_j)\right]. \tag{33}$$

Also note that in the Bayes SPECP's for the string and tree languages proposed previously in Section IV-C and IV-D, only the minimum Bayes distance

$$B(\omega', C_i) = \min_{\substack{j=1,2,\cdots,k_i \\ \omega_j \in C_i}} B(\omega', \omega_j) \tag{34}$$

is computed. So, in order to compute the conditional probability or density function

$$p(\omega'|C_i) = \sum_{\substack{j=1 \\ \omega_j \in C_i}}^{k_i} \exp\left[-B(\omega', \omega_j)\right], \tag{35}$$

the two Bayes SPECP algorithms (Algorithm 1 and 2) must be modified. This is discussed in the next section. We call a classification scheme using the above optimum multi-class classification rule (32) a *Bayes error-correcting recognition (BECR) system.*

B. Modification of Bayes SPECP's for Bayes Error-Correcting Recognition System

Bayes SPECP's which are useful for *intraclass* pattern classification only are modified in this section to serve the purpose of *interclass* Bayes error-correcting recognition. Since we want to obtain the Bayes distances between an input observed pattern ω' and all the pure patterns from which ω' may be deformed statistically, the modification is made such that all possible partial derivations of ω', instead of only the one with a minimum *partial distance* (d_{ih}, e_{ih} in Algorithm 1, or d, d_0 in Algorithm 2), are kept in the intermediate steps. As a distinction, the resulting SPECP's are called *interclass SPECP's.*

Algorithm 3: Interclass SPECP for string languages for BECR.

Input: Same as that of Algorithm 1.

Output: A set of pure string representations $\{x_1, x_2, \cdots, x_L\}$ accepted by G_s with a set of Bayes distances $B(y, x_i)$, $B(y, x_2), \cdots, B(y, x_L)$, if y is structure-preserved.

Method: Same as that of Algorithm 1 except

1) In Step 1, set
$$t_{i1} = \{(A, d_{ih}, k_h) \mid d_{ij} \neq \infty,\ h = 1, 2, \cdots, n_A,\ A \in V_N\}.$$
2) In Step 2, set
$$t_{ij} = \{(A, e_{ih}, k_h) \mid h = 1, 2, \cdots, n'_A,\ A \in V_N\}.$$
3) In Step 4, when examining t_{1n}, let L be the total number of triplets in t_{1n}, each of the form (S, d, k) for some d and k. Then for ith such triplet (S, d_i, k_i), set $B(y, x_i) = d_i$. If $L = 0$, then y is not structure-preserved.

Algorithm 4: Interclass SPECP for tree languages for BECR.

Input: Same as that of Algorithm 2.

Output: A set of pure tree representations $\{\alpha_1, \alpha_2, \cdots, \alpha_L\}$ accepted by G_s with a set of Bayes distances $B(\beta, \alpha_1)$, $B(\beta, \alpha_2), \cdots, B(\beta, \alpha_L)$, if β is structure-preserved.

Method: Same as that of Algorithm 2 except

1) Step 3 should be deleted.
2) In Step 5, when examining t_0, let L be the total number of triplets in t_{1n}, each of the form (S, d, k) for some d and k. Then for ith such triplet (S, d_i, k_i), set $B(\beta, \alpha_i) = d_i$. If $L = 0$, then y is not structure-preserved.

C. A Suboptimal Bayes Error-Correcting Recognition System

The Bayes error-correcting recognition system proposed previously, though optimum statistically, is impractical when the pattern grammar used is highly ambiguous, since the parsing will then become very inefficient due to the accumulation of many triplets (A, d, k) in the entries t_{ij} of the parse table (Algorithm 3), or in the entries t_b of the transition table (Algorithm 4). Note that each such triplet corresponds to a partial derivation of the input pattern representation. So, in practical applications, the pattern grammar usually is made unambiguous.[4] Also, we assume that each observed pattern is deformed from only one pure pattern. Then, without loss of generality, we can infer a pattern grammar G_i for pattern class C_i in such a way that even though an input observed pattern ω' *grammatically* may have several error-correcting parses $\omega_1, \omega_2, \cdots, \omega_{k_i}$ with respect to G_i, *statistically* only one of these k_i pure patterns will result in a large value of $p(\omega' \mid \omega_j)P(\omega_j \mid C_i)$ being computed, compared with those of other remaining ω_j. So, in a suboptimal sense, the conditional probability or density function $p(\omega' \mid C_i)$ can be approximated by

$$p(\omega' \mid C_i) = \sum_{j=1}^{k_i} p(\omega' \mid \omega_j)P(\omega_j \mid C_i) \simeq \max_{j=1,2,\cdots,k_i} p(\omega' \mid \omega_j)P(\omega_j \mid C_i) \qquad (36)$$

which, by using the output minimum Bayes distance $B(\omega', C_i)$ of the more efficient Bayes SPECP (Algorithm 1 or 2), can be computed as

$$p(\omega' \mid C_i) = \exp\left[-B(\omega', C_i)\right].$$

We thus have a more efficient, though suboptimal, error-correcting recognition rule for practical applications [10]. That is, decide $\omega' \sim C_l$ if

$$\left[\max_{\substack{j=1,2,\cdots,k_l\\ \omega_j \in C_l}} p(\omega' \mid \omega_j)P(\omega_j \mid C_l)\right] P(C_l) = \max_{i=1,2,\cdots,m}\left[\max_{\substack{j=1,2,\cdots,k_i\\ \omega_j \in C_i}} p(\omega' \mid \omega_j)P(\omega_j \mid C_i)\right] P(C_i). \qquad (37)$$

A recognition system using such a decision rule is called a *suboptimal Bayes error-correcting recognition system.*

The above suboptimal recognition system essentially has also been proposed by Fung and Fu [18] and Lu and Fu [20] for syntactic local deformations. The proposed Bayes error-correction recognition system (Section V-A) and its suboptimal version, however, not only can perform stochastic syntax analysis of input pattern structures by using the SPECP's, but also can take the numerical information contained in pattern primitives into consideration. Therefore, they can be regarded as hybrid pattern classifiers because advantages of both syntactic and statistical pattern recognition techniques have been utilized.

[4] Note that unambiguity of the pattern grammar does not guarantee a unique error-correcting parse of an input pattern.

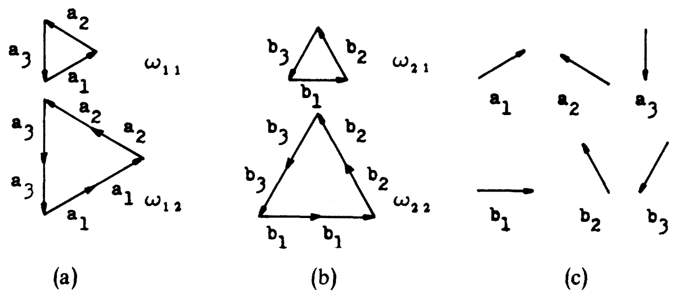

Fig. 4. (a) Pure pattern class C_1. (b) Pure pattern class C_2. (c) Pure primitives for Fig. 4(a) & (b).

Compared with the syntactic recognition approach using stochastic grammars only [7], [15], the proposed deformational scheme may be regarded as a special case of *stochastic transformational grammar* which is expected to handle complex noisy input patterns when simple stochastic grammars are not adequate to apply [3].

D. An Illustrative Example—Classification using the BECR System

An example for string languages is given in this section to illustrate the applicability of the proposed (optimum) Bayes error-correcting recognition system and to compare its performance with other error-correcting systems which handle continuous semantic information by thresholding it into finite discrete levels.

Example 2) Classification of a deformed triangle: Assume that we have two pure pattern classes. One pattern class C_1 consists of two equilateral triangles ω_{11}, ω_{12}, as shown in Fig. 4(a), and the other class C_2 consists of two other different equilateral triangles ω_{21}, ω_{22} as shown in Fig. 4(b). The primitives used, which are fixed-length line segments, are shown in Fig. 4(c).

Also assume the following probability values: $P(C_1) = 0.5$, $P(C_2) = 0.5$, $P(\omega_{11} \mid C_1) = 0.60$, $P(\omega_{12} \mid C_1) = 0.40$, $P(\omega_{21} \mid C_2) = 0.80$, $P(\omega_{22} \mid C_2) = 0.20$. Two stochastic pattern grammars G_1, G_2, consistent with these probabilities for C_1, C_2, respectively, are given in the following:

$$G_1 = (V_{N1}, V_{T1}, P_1, S_1)$$

$$V_{N1} = \{A, B, C, D, A_1, B_1, C_1, D_1\}$$

$$V_{T1} = \{a_1, a_2, a_3\}$$

P_1:
1) $S_1 \xrightarrow{0.6} AD$
2) $S_1 \xrightarrow{0.4} A_1D_1$
3) $D \xrightarrow{1.0} BC$
4) $A_1 \xrightarrow{1.0} AA$
5) $D_1 \xrightarrow{1.0} B_1C_1$

6) $B_1 \xrightarrow{1.0} BB$

7) $C_1 \xrightarrow{1.0} CC$

8) $A \xrightarrow{1.0} a_1$

9) $B \xrightarrow{1.0} a_2$

10) $C \xrightarrow{1.0} a_3$

and

$$G_2 = (V_{N2}, V_{T2}, P_2, S_2)$$
$$V_{N2} = \{A, B, C, D, A_1, B_1, C_1, D_1\}$$
$$V_{T2} = \{b_1, b_2, b_3\}$$

P_2: 1) $S_2 \xrightarrow{0.8} AD$

2) $S_2 \xrightarrow{0.2} A_1 D_1$

3) $D \xrightarrow{1.0} BC$

4) $A_1 \xrightarrow{1.0} AA$

5) $D_1 \xrightarrow{1.0} B_1 C_1$

6) $B_1 \xrightarrow{1.0} BB$

7) $C_1 \xrightarrow{1.0} CC$

8) $A \xrightarrow{1.0} b_1$

9) $B \xrightarrow{1.0} b_2$

10) $C \xrightarrow{1.0} b_3$.

To use the interclass SPECP of Algorithm 3 for illustrative purpose, the above two grammars are inferred in their context-free forms, although simpler finite-state grammars can certainly be used. They are also in Chomsky normal form.

Now assume that each pattern ω_{ij} ($i = 1, 2, j = 1, 2$) is subject to both syntactic and semantic local deformations with each line segment in ω_{ij} being deformed independently. Each line segment can be syntactically deformed into a curve segment with a fixed curvature and a fixed length but with a variable direction. We use the 2-tuple (L, θ) and (C, θ) to characterize the pure primitives—line segments, and the deformed primitives—curve segments, respectively, where L and C are syntactic symbols, and θ denotes the one-dimensional semantic vector—the direction of the primitives with respect to x-axis. So we have all the 2-tuples for the pure primitives shown in Fig. 4(c) as

$$a_1 = (L, 30°) \qquad b_1 = (L, 0°)$$
$$a_2 = (L, 150°) \qquad b_2 = (L, 120°)$$
$$a_3 = (L, 270°) \qquad b_3 = (L, 240°).$$

We also assume that each a_i ($i = 1, 2, 3$) can be deformed syntactically into a curve segment with probability 0.1, and that each b_i ($i = 1, 2, 3$) can be deformed syntactically into a curve segment with probability 0.13. Furthermore, each line or curve segment is semantically deformed on its direction θ approximately with a normal distribution as shown in the following data (for notations, see Section III-B):

$$D_{a_i} = \{a_{i1} = a_i = (L, \theta_{a_i}), a_{i2} = (C, \theta_{a_i})\}$$

where

$$\theta_{a_i} = 30° + (i - 1) \cdot 120°$$

with

$$p(a_{i1} \mid a_i) = 0.9, \; p(a_{i2} \mid a_i) = 0.1, \qquad \text{for } i = 1, 2, 3.$$

$$D_{b_i} = \{b_{i1} = b_i = (L, \theta_{b_i}), b_{i2} = (C, \theta_{b_i})\}$$

where

$$\theta_{b_i} = (i - 1) \cdot 120°$$

with[5]

$$p(b_{i1} \mid b_i) = 0.87, \; p(b_{i1} \mid b_i) = 0.13, \qquad \text{for } i = 1, 2, 3.$$

$$D_{a_{ij}} = \{a_{ijk} \mid a_{ijk} = (S_j, \theta_k), \; |\theta_k - \theta_{a_i}| \le 40°\}$$

where

$$i = 1, 2, 3, \qquad j = 1, 2,$$
$$S_j = L, \qquad \text{when } j = 1$$
$$= C, \qquad \text{when } j = 2,$$

and

$$q(a_{ijk} \mid a_{ij}, a_i) = \frac{1}{\sqrt{2\pi}\,\sigma_a} \exp\left[-\tfrac{1}{2}(\theta_k - \theta_{a_i})^2/\sigma_a^2\right]$$

with

$$\sigma_a = 8°, \; \theta_{a_i} = 30° + (i - 1) \cdot 120°,$$

$$D_{b_{ij}} = \{b_{ijk} \mid b_{ijk} = (S_j, \theta_k), \; |\theta_k - \theta_{b_i}| \le 40°\}^5$$

where

$$i = 1, 2, 3, \qquad j = 1, 2,$$
$$S_j = L, \qquad \text{when } j = 1$$
$$= C, \qquad \text{when } j = 2,$$

and

$$q(b_{ijk} \mid b_{ij}, b_i) = \frac{1}{\sqrt{2\pi}\,\sigma_b} \exp\left[-\tfrac{1}{2}(\theta_k - \theta_{b_i})^2/\sigma_b^2\right]$$

with

$$\sigma_b = 10°, \; \theta_{b_i} = (i - 1) \cdot 120°.$$

The six semi-pure primitives, i.e., the six curve segments corresponding to a_{12}, a_{22}, a_{32}, and b_{12}, b_{22}, b_{32} are shown in Fig. 5(a). Two possible observed patterns are shown in Fig. 5(b) and Fig. 5(c), respectively.

[5] Mathematically, there is no limitation on the value of θ_k, so the assumption is strictly for computational convenience.

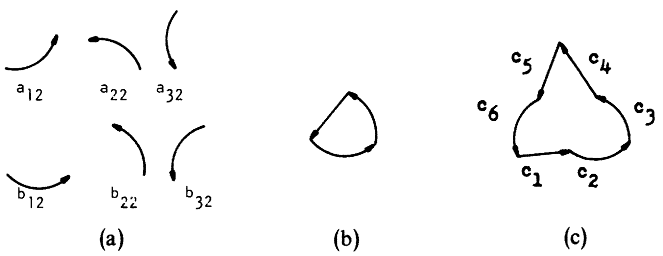

Fig. 5. (*a*) Semi-pure primitives for Fig. 4(a) and (b). (b) An observed pattern. (c) Another observed pattern ω'

$(S_1,36.68,2)$					
ϕ	ϕ				
ϕ	ϕ	$(D_1,23.84,5)$			
ϕ	ϕ	ϕ	ϕ		
$(A_1,11.92,4)$	ϕ	$(B_1,11.92,6)$	ϕ	$(C_1,11.92,7)$	
$(A,4.86,8)$	$(A,7.06,8)$	$(B,7.06,9)$	$(B,4.86,9)$	$(C,4.86,10)$	$(C,7.06,10)$

(a)

$(S_2,34.19,2)$					
ϕ	ϕ				
ϕ	ϕ	$(D_1,21.72,5)$			
ϕ	ϕ	ϕ	ϕ		
$(A_1,10.86,4)$	ϕ	$(B_1,10.86,6)$	ϕ	$(C_1,10.86,7)$	
$(A,4.48,8)$	$(A,6.38,8)$	$(B,6.38,9)$	$(B,4.48,9)$	$(C,4.48,10)$	$(C,6.38,10)$

(b)

Fig. 6. (a) Parse table T_1. (b) Parse table T_2.

Now suppose we want to classify the deformed pattern ω' shown in Fig. 5(c) with the following string representation:

$$\omega' = c_1 c_2 c_3 c_4 c_5 c_6$$

where

$$c_1 = (L, 15°), \qquad c_4 = (L, 135°),$$
$$c_2 = (C, 15°), \qquad c_5 = (L, 255°),$$
$$c_3 = (C, 135°), \qquad c_6 = (C, 255°).$$

To apply the BECR system, at first, we use the interclass SPECP's (Algorithm 3) for grammar G_1 and G_2 to parse ω'. We obtain two parse tables T_1, T_2 for G_1 and G_2, respectively, as shown in Fig. 6(a) and (b). Since S_1 is in t_{16} of T_1, and S_2 in t_{16} of T_2, ω' is accepted by both classes C_1 and C_2 with Bayes distances $d_1 = 36.68$ and $d_2 = 34.19$, respectively. Since only one triplet exists in each t_{16}, using Bayes SPECP's will also get the same result. Next, we apply the interclass Bayes decision rule, compute

$$\begin{aligned} P(C_1 \mid \omega') &= p(\omega' \mid C_1)P(C_1) \\ &= \exp(-36.68) \cdot 0.5 \\ &= 5.88 \times 10^{-17} \end{aligned}$$

$$\begin{aligned} P(C_2 \mid \omega') &= \exp(-34.19) \cdot 0.5 \\ &= 70.87 \times 10^{-17}, \end{aligned}$$

and decide that ω' belongs to C_2. This completes our illustrative example for the proposed Bayes error-correcting recognition system.

In the following, we threshold the continuous θ values into intervals as is usually done in other error-correcting schemes, and show how contrary decision can be made for the previous input pattern ω'. Since the proposed Bayes recognition system always gives optimum decisions in the Bayes sense, we thus have shown its better performance than other systems using thresholding approaches on continuous semantic information.

If we threshold θ values starting from 0°[6] in steps of 20° for class C_1, and from 30°[6] in steps of 20° for C_2, then $D_{a_{ij}}$ and $D_{b_{ij}}$ can be changed to the following:

$$D_{a_{ij}} = \{a_{ijk} \mid k = 1, 2, 3, 4,\ a_{ijk} = (S_j, \theta_k),\ (k-2) \cdot 20° \le \theta_k - \theta_{a_i} \le (k-1) \cdot 20°\}$$

with discrete probabilities

$$q(a_{ijk} \mid a_{ij}, a_i) = \begin{cases} 0.01, & k = 1, 4 \\ 0.49, & k = 2, 3, \end{cases}$$

$$D_{b_{ij}} = \{b_{ijk} \mid k = 1, 2, 3, 4,\ b_{ijk} = (S_j, \theta_k),\ (k-2) \cdot 20° \le \theta_k - \theta_{b_i} \le (k-1) \cdot 20°\}$$

with discrete probabilities

$$q(b_{ijk} \mid b_{ij}, b_i) = \begin{cases} 0.02, & k = 1, 4 \\ 0.48, & k = 2, 3, \end{cases}$$

with S_j the same as defined previously. And by convention, only the following probability values are used in parsing [3]:

$$\begin{aligned} &r(a_{ijk} \mid a_i) \\ &\quad = q(a_{ijk} \mid a_i, a_{ij}) \cdot p(a_{ij} \mid a_i) = \begin{cases} 0.009, & j = 1, k = 1, 4 \\ 0.441, & j = 1, k = 2, 3 \\ 0.001, & j = 2, k = 1, 4 \\ 0.049, & j = 2, k = 2, 3 \end{cases} \end{aligned}$$

$$\begin{aligned} &r(b_{ijk} \mid b_i) \\ &\quad = q(b_{ijk} \mid b_i, b_{ij}) \cdot p(b_{ij} \mid b_i) = \begin{cases} 0.0174, & j = 1, k = 1, 4 \\ 0.4176, & j = 1, k = 2, 3 \\ 0.0026, & j = 2, k = 1, 4 \\ 0.0624, & j = 2, k = 2, 3 \end{cases} \end{aligned}$$

$i = 1, 2, 3$. The previous data show that each a_i or b_i can be deformed into eight different observed primitives with different probabilities, in which four are line segments and the other four are curve segments.

Now again use the interclass SPECP's (Algorithm 3) for G_1, G_2 to parse ω', respectively. Note that after thresholding the θ values in ω' and transforming into string representa-

[6] Starting from different points to threshold is just for convenience because the directions of a_1, b_1 are 0° and 30°.

$(S_1,12.44,2)$					
ϕ	ϕ				
ϕ	ϕ	$(D_1,7.68,5)$			
ϕ	ϕ	ϕ	ϕ		
$(A_1,3.84,4)$	ϕ	$(B_1,3.84,6)$	ϕ	$(C_1,3.84,7)$	
$(A,0.82,8)$	$(A,3.02,8)$	$(B,3.02,9)$	$(B,0.82,9)$	$(C,0.82,10)$	$(C,3.02,10)$

(a)

$(S_2,12.53,2)$					
ϕ	ϕ				
ϕ	ϕ	$(D_1,7.28,5)$			
ϕ	ϕ	ϕ	ϕ		
$(A_1,3.64,4)$	ϕ	$(B_1,3.64,6)$	ϕ	$(C_1,3.64,7)$	
$(A,0.87,8)$	$(A,2.77,8)$	$(B,2.77,9)$	$(B,0.87,9)$	$(C,0.87,10)$	$(C,2.77,10)$

(b)

Fig. 7. (a) Parse table T_3. (b) Parse table T_4.

tions, we get

$$\omega' = a_{113}a_{123}a_{223}a_{213}a_{313}a_{323}$$

for class C_1, or

$$\omega' = b_{112}b_{122}b_{222}b_{212}b_{312}b_{322}$$

for class C_2. Also note that the term $[\ln p(s_i|t_h) + \ln q(x_i|t_h, s_i)]$ contained in d_{ih} of Algorithm 3 should be replaced by $\ln r(c_i|a_h)$ before the algorithm is applied to our discrete case here, where $c_i = a_{ijk}$ or b_{ijk} now. From the resulting parse tables (Fig. 7(a) and (b)), we get

$$P(C_1|\omega') = \exp(-12.44)\cdot 0.5 = 1.98 \times 10^{-6}$$

$$P(C_2|\omega') = \exp(-12.53)\cdot 0.5 = 1.81 \times 10^{-6}$$

and decide that ω' belongs to C_1!

A careful study reveals that such contrary conclusion to the previous Bayesian decision $\omega' \sim C_2$ is due to the coarse thresholding used. Using smaller intervals in thresholding will improve the result, but will not be better than that obtained from the proposed system.

VI. Concluding Remarks

Suboptimal Bayes error-correcting recognition systems using Bayes error-correcting parsers and Bayes interclass decision rule have been proposed both by Fung and Fu [18] and by Lu and Fu [20]. The proposed systems described in this paper can be considered, from the viewpoint of local deformation, as a generalization of theirs with respect to the use of semantic information, which is often more relevant for practical pattern recognition when both structural and numerical informations are available for primitive discrimination [6], [13], [19]. Further investigations should be directed to include error-correcting capability for structural deformations under the formalism of the proposed deformational model and thus provide a complete error-correcting recognition system.

References

[1] A. V. Aho and T. G. Peterson, "A minimum distance error-correcting parser for context-free languages," *SIAM J. Comput.*, vol. 1, pp. 305–312, Dec. 1972.

[2] R. A. Thompson, "Language correction using probabilistic grammars," *IEEE Trans. on Comput.*, vol. C-25, no. 3, Mar. 1976.

[3] L. W. Fung and K. S. Fu, "Stochastic syntactic decoding for pattern classification," *IEEE Trans. on Comput.*, vol. C-24, no. 6, July 1975.

[4] S. Y. Lu and K. S. Fu, "Structure-preserved error-correcting tree automata for syntactic pattern recognition," presented at IEEE Conf. on Decision and Control, Clearwater Beach, FL, Dec. 1976.

[5] W. H. Tsai and K. S. Fu, "Image segmentation and recognition by texture discrimination: A syntactic approach," presented at Proc. 4th Int. Joint Conf. on Pattern Recognition, Kyoto, Japan, Nov. 1978.

[6] C. H. Chen, "On statistical and structural feature extraction," presented at Joint Workshop on Pattern Recognition and Artificial Intelligence, Hyannis, MA, June 1976.

[7] K. S. Fu, *Syntactic Methods in Pattern Recognition*. New York: Academic, 1974.

[8] R. Narasimhan, "On the description, generation, and recognition of classes of pictures," in *Automatic Interpretation and Classification of Images*, A. Grasselli, Ed. New York: Academic, 1969.

[9] K. C. You and K. S. Fu, "Syntactic shape recognition using attributed grammars," *IEEE Trans. on Syst., Man, Cybern.*, vol. SMC-9, no. 6, June 1979.

[10] W. H. Tsai and K. S. Fu, "A pattern deformational model and Bayes error-correcting recognition system," presented at Proc. Int. Conf. on Cybern. and Society, Tokyo, Japan, Nov. 1978.

[11] S. Y. Lu and K. S. Fu, "Error-correcting tree automata for syntactic pattern recognition," presented at Proc. 1978 IEEE Conf. on Pattern Recognition and Image Processing, Troy, NY, 1978.

[12] B. K. Bhargava and K. S. Fu, "Transformation and inference of grammars for syntactic pattern recognition," presented at Proc. 1974 IEEE Int. Conf. on Cybern. and Society, Dallas, TX, Oct. 1974.

[13] U. Grenander, "A unified approach to pattern analysis," in *Advances in Computers*, Vol. 10. New York: Academic, 1970.

[14] V. A. Kovalevsky, "Sequential optimization in pattern recognition and pattern description," in *Proc. Int. Fed. Information Processing Congr.*, Amsterdam, the Netherlands, 1968.

[15] H. C. Lee and K. S. Fu, "A syntactic pattern recognition system with learning capability," presented at Proc. Int. Symposium on Comput. and Inform. Science, 4th Miami Beach, FL, Dec. 1972.

[16] A. V. Aho and J. D. Ullman, *The Theory of Parsing, Translation, and Compiling, Vol. 1*. Englewood Cliffs, NJ: Prentice-Hall, 1972.

[17] K. S. Fu and B. K. Bhargava, "Tree systems for syntactic pattern recognition," *IEEE Trans. on Comput.*, vol. C-22, no. 12, Dec. 1973.

[18] L. W. Fung and K. S. Fu, "Syntactic decoding for computer communication and pattern recognition," Purdue Univ., TR-EE 74-47, Dec. 1974.

[19] L. Kanal and B. Chandrasekaran, "On linguistic, statistical, and mixed models for pattern recognition," in *Frontier of Pattern Recognition*, S. Watanabe, Ed. New York: Academic, 1972.

[20] S. Y. Lu and K. S. Fu, "Stochastic error-correcting syntax analysis for recognition of noisy patterns," *IEEE Trans. on Comput.*, vol. C-26, no. 12, Dec. 1977.

[21] A. C. Shaw, "A formal picture description scheme as a basis for picture processing systems," *Inform. and Control*, vol. 14, pp. 9–52, 1969.