Modified Quadratic Discriminant Functions and the Application to Chinese Character Recognition

FUMITAKA KIMURA, KENJI TAKASHINA, SHINJI TSURUOKA, AND YASUJI MIYAKE

Abstract—**Issues in the quadratic discriminant functions (QDF) are discussed and two types of modified quadratic disriminant functions (MQDF1, MQDF2) which are less sensitive to the estimation error of the covariance matrices are proposed. The MQDF1 is a function which employs a kind of a (pseudo) Bayesian estimate of the covariance matrix instead of the maximum likelihood estimate ordinarily used in the QDF. The MQDF2 is a variation of the MQDF1 to save the required computation time and storage.**

Two discriminant functions were applied to Chinese character recognition to evaluate their effectiveness, and remarkable improvement was observed in their performance.

Index Terms—**Bayes classifier, character recognition, Chinese character, OCR, parameter estimation, peaking phenomenon, quadratic discriminant function, statistical pattern recognition.**

I. Introduction

A quadratic discriminant function (QDF) of an n-dimensional feature vector is given as

$$g_0^{(l)}(x) = (x - \mu^{(l)})^t \{\Sigma^{(l)}\}^{-1} (x - \mu^{(l)}) + \log |\Sigma^{(l)}| - 2 \log P(\omega^{(l)}) \quad (1)$$

for a class $\omega^{(l)}$ where $\mu^{(l)}$ and $\Sigma^{(l)}$ denote the mean vector and the convariance matrix for x in the class $\omega^{(l)}$, respectively, and $P(\omega^{(l)})$ is the *a priori* probability for the class $\omega^{(l)}$. The QDF becomes optimal in the Bayesian sense for normal distributions with known parameters. In this case, the QDF has the following properties: 1) optimality—the QDF achieves the minimum mean error probability, and 2) monotonicity—the average error rate[1] of the QDF decreases monotonically with an increase of the feature size [1], [2].

In this paper, we assumed that parameters of relating distributions are unknown. In such a case, the maximum likelihood estimates are usually used as the parameters in the QDF. Hereafter, we shall call the QDF with the maximum likelihood estimates simply QDF.

The QDF suffers from the following issues.

1) The performance is degraded because of estimation errors in the parameters. It becomes nonoptimal, that is, it cannot achieve the minimum error rate. It also loses the monotonicity described in 2) above and shows the so-called peaking phenomenon: the average error rate decreases with addition of new features until the feature size reaches a certain optimal size. After that, it again increases with an increase in the number of features. Thus, the performance becomes poorer if the feature size continues to increase over the appropriate feature size giving the peak in recognition probability.

2) It requires much computation time and storage. Since the QDF employs the covariance matrix, required computation time and storage is $O(n^2)$ for n-dimensional feature vectors, which is impractical for n greater than 20 or 30.

Manuscript received October 19, 1984; revised April 25, 1986. Recommended for acceptance by J. Kittler.

The authors are with the Department of Electronics, Mie University, Kamihama-cho 1515, Tsu 514, Japan.

IEEE Log Number 8609969.

[1]The term average error rate means the error probability averaged with regard to the parameters of the distributions.

3) The effect of the deviation from the normal distribution is unknown. However, we do not discuss this issue in the paper.

There are several approaches for the first issue.

1) Feature Selection: Feature selection techniques such as the principal component analysis and the sequential feature selection [3] are employed to reduce the feature size. This approach is effective to reduce the error rate when the original feature size is so large that the performance is degraded by the peaking phenomenon. But it is not suitable if the feature size necessary to separate patterns is intrinsically large because separability in the feature space is also reduced by feature selection, although the estimation errors of the parameters can be reduced.

2) Alternative Estimates of Parameters: Sometimes, ad hoc estimates or those which are obtained by using prior knowledge about the parameters involve fewer estimation errors and give better performance than the maximum likelihood estimates [4], [5]. For example, if it is known that the covariance matrices for all classes are identical, the estimation error may be reduced by pooling whole samples to estimate the covariance matrix. If features are known to be mutually independent, nondiagonal elements of the covariance matrix should be set to zero, no matter what their maximum likelihood estimates are.

In this paper, the second approach is adopted, that is, sensitivity of the QDF to the estimation error of the covariance matrix is discussed and a modified quadratic discriminant function (MQDF1) less sensitive to the error is proposed. A variation of the MQDF1, called MQDF2, is also proposed, which requires less computation time and storage than the QDF and the MQDF1.

For higher dimensional patterns, nonparametric techniques perform well compared to the QDF, even for the Gaussian problems [6]. The reason is that the nonparametric technique need not use the covariance matrix and is not affected by the estimation error. However, the nonparametric technique is not appropriate to Gaussian problems because

a) it requires computation cost and storage proportional to the number of learning samples, and

b) it does not explicitly utilize the prior knowledge that the underlying distribution is Gaussian.

These defects can be overcome by the MQDF1 which is led by parametrically approximating the distribution estimated by a nonparametric technique. The relationship between the MQDF1 and the nonparametric technique is described in Section II.

II. Modified Quadratic Discriminant Functions

The QDF can be written as

$$g_0(x) = \sum_{i=1}^{n} \frac{1}{\lambda_i} \{\varphi_i^t(x - \mu_M)\}^2 + \log \prod_{i=1}^{n} \lambda_i \quad (2)$$

by using the equation

$$\Sigma_M = \sum_{i=1}^{n} \lambda_i \varphi_i \varphi_i^t \quad (3)$$

where μ_M and Σ_M denote the maximum likelihood estimates of the mean and the covariance, respectively, and $\lambda_i (\lambda_i \geqq \lambda_{i+1})$ and φ_i denote the ith eigenvalue and the eigenvector of the matrix Σ_M. For simplicity, the subscript l representing the class and the term of $P(\omega^{(l)})$ is omitted. This orthogonal expansion form of the QDF plays a very important role in seeing the effect of the estimation error in the covariance matrix of the QDF and in improving it as described below. The effect of the estimation error in the covariance matrix is evaluated by estimating those in eigenvalues and eigenvectors because of the one-to-one correspondence between a covariance matrix and a set of eigenvalues and eigenvectors. The important point here is the fact that the estimation errors in the nondominant eigenvectors are much greater than those of the dominant eigenvectors [7]. This means that the nondominant eigenvector terms (terms for larger i's) in (2) are much more sensitive to the estimation error in the covariance matrix and may critically degrade the performance of the QDF. This phenomenon is discussed in Section III with experimental results.

We do not take into consideration the estimation error in the mean vector of the QDF because it is expected that its effect is not as serious as that of the covariance matrix [8].

Performance of the discriminant function is fairly improved by employing a kind of pseudo-Bayesian estimate [4] of the covariance matrix

$$\Sigma_P = \Sigma_M + h^2 I \tag{4}$$

instead of the maximum likelihood estimate Σ_M in the QDF. Here, I is the (n-dimensional) identity matrix and h^2 is an appropriate constant.

From (4),

$$\varphi_i \Sigma_P \varphi_i^t = \varphi_i \Sigma_M \varphi_i^t + h^2 = \lambda_i + h^2 \tag{5}$$

where λ_i and φ_i are the ith eigenvalue and eigenvector of Σ_M. Therefore, the ith eigenvalue and eigenvector of Σ_P are equal to $\lambda_i + h^2$ and φ_i, respectively. Thus, an improved discriminant function MQDF1 is given as

$$g_1(x) = \sum_{i=1}^{n} \frac{1}{\lambda_i + h^2} \{\varphi_i^t(x - \mu_M)\}^2 + \log \prod_{i=1}^{n} (\lambda_i + h^2). \tag{6}$$

It has been pointed out that various estimates other than the maximum likelihood estimates sometimes give better results [4], [5].

The MQDF1 can also be considered as a function derived by substituting the covariance matrix Σ_M in the QDF by

$$\Sigma_{P'} = \int (x - \mu_M)(x - \mu_M)^t \, p_m(x) \, dx, \tag{7}$$

which is the covariance matrix of the density $p_m(x)$ estimated by the Parzen-window approach using circular normal density as the kernel. The Parzen estimate $p_m(x)$ of the density and the kernel K are given as

$$p_m(x) = \frac{1}{m} \sum_{j=1}^{m} (1/h^n) \, K[(x - x_j)/h]$$

$$(1/h^n) \, K[(x - x_j)/h] = (2\pi)^{-n/2} h^{-n} \times \exp\left[-\tfrac{1}{2} h^{-2} (x - x_j)^t (x - x_j)\right] \tag{8}$$

where m, h^2, and x_j denote the sample size, the variance of the kernel, and the jth sample, respectively [7]. From (7) and (8),

$$\Sigma_{P'} = \frac{1}{m} \sum_{j=1}^{m} (x_j - \mu_M)(x_j - \mu_M)^t + h^2 I = \Sigma_M + h^2 I \tag{9}$$

The matrix $\Sigma_{P'}$ is no more than the maximum likelihood estimate of the covariance matrix with h^2 added to its diagonal elements. The MQDF1 becomes identical to the QDF when h^2 is set to zero. The covariance matrix $\Sigma_{P'}$ as well as the density $p_m(x)$ is asymptotically unbiased, asymptotically consistent, and uniformly consistent under some conditions on h^2. The conditions and an example of the h^2 are described in [7]. The larger the number of samples m and the smaller the feature size n, the smaller the value that should be selected as the constant h^2. Both required computation time and storage of MQDF1 are $O(n^2)$, which is the same as those of the QDF.

Another modification of the discriminant function MQDF2 is derived as follows by substituting h^2 for all of the eigenvalues λ_i, $i \geqq k + 1$ of Σ_M in QDF.

$$g_2(x) = \sum_{i=1}^{k} \frac{1}{\lambda_i} \{\varphi_i^t(x - \mu_M)\}^2 + \sum_{i=k+1}^{n} \frac{1}{h^2} \{\varphi_i^t(x - \mu_M)\}^2 + \log\left(h^{2(n-k)} \prod_{i=1}^{k} \lambda_i\right). \tag{10}$$

This is equal to the function obtained by neglecting h^2 for $i \leqq k$ and λ_i for $k < i \leqq n$ in (6). By using the equation

$$\sum_{i=1}^{n} \{\varphi_i^t(x - \mu_M)\}^2 = \|x - \mu_M\|^2, \tag{11}$$

the MQDF2 of (10) is rewritten as

$$g_2(x) = \frac{1}{h^2}\left[\|x - \mu_M\|^2 - \sum_{i=1}^{k}\left(1 - \frac{h^2}{\lambda_i}\right)\{\varphi_i^t(x - \mu_M)\}^2\right] + \log\left(h^{2(n-k)} \prod_{i=1}^{k} \lambda_i\right). \tag{12}$$

From (12), it is obvious that the required computation time and storage of the MQDF2 are about k/n times those of the QDF and the MQDF1. The value of k should be selected so that the estimation error of λ_i, φ_i ($i = 1, \cdots, k$) does not become too large. In practice, k is regarded as a constant which does not depend on n because it is extremely difficult to increase the number of learning samples and thus the value of k with an increase in the feature size n. Accordingly, the required computation time and storage of MQDF2 are both $O(n)$.

It is also expected that the MQDF2 as well as the MQDF1 are less sensitive to the estimation error of the covariance matrix if h^2 and k are suitably chosen. We will choose the values by experiment in Section III. The MQDF2 is reduced to the QDF for $k = n$ and to the Euclidean distance for $k = 0$, respectively. If $0 < k < n$, the MQDF2 consists of the weighted sum of the QDF in the k-dimensional subspace ${}_1S_k$ spanned by φ_i ($i = 1, \cdots, k$), and the Euclidean distance in the ($n - k$)-dimensional subspace ${}_{k+1}S_n$ spanned by φ_i ($i = k + 1, \cdots, n$). The reciprocal of the variance $1/h^2$ corresponds to the weight. The estimate of the covariance matrix empolyed in MQDF2 is given as

$$\sum_{i=1}^{k} \lambda_i \varphi_i \varphi_i^t + h^2 \sum_{i=k+1}^{n} \varphi_i \varphi_i^t = \sum_{i=1}^{k} (\lambda_i - h^2) \, \varphi_i \varphi_i^t + h^2 I. \tag{13}$$

Compared to (4), this estimate can be regarded as a kind of pseudo-Bayesian estimate.

III. Experiments

We carried out Experiment I to show that the nondominant eigenvector terms in (2) are more sensitive to the estimation error of the covariance and may degrade the performance of the QDF. We also conducted Experiment II; 1) to know the relationship between h^2 and the classification rate for the MQDF1, 2) to examine the relationship among h^2, k, and the classification rate for the MQDF2 for finding the proper value of h^2 and k to achieve the best performance, and 3) to find the classification rate for the QDF, and the MQDF1, the MQDF2, the Euclidean distance, the city-block distance, and the linear discriminant function and to compare their performances.

The pooled-within covariance matrix was employed as a covariance matrix for the linear discrimant function.

For the MQDF2, we should evaluate the performance varying both h^2 and k independently. However, we set the value of h^2 equal to the average ($\overline{\lambda}_{k+1}$) of λ_{k+1} over all classes, and varied only the value of k to reduce the cost of the experiment. Accordingly, h^2

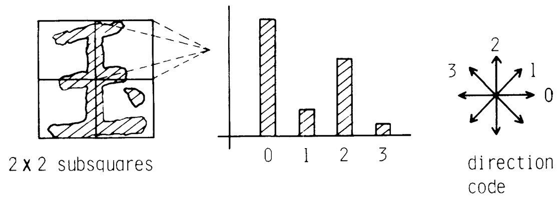

Fig. 1. Histogram of the direction code.

TABLE I
CLASSIFICATION RATES (PERCENT)

	Euclidean distance	QDF	MQDF2 (k=20)
learning samples (Vol.8)	89.5	100.0	100.0
test samples (Vol.6)	85.2	93.1	98.4

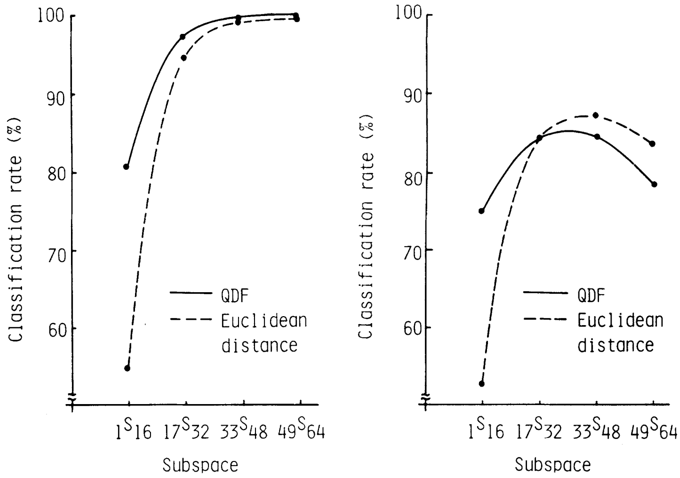

Fig. 2. Classification rates in subspaces ${}_kS_{k+15}$: (a) for learning samples (ETL8 database, vol. 8), (b) for test samples (ETL8 database, vol. 6).

depends on k in such a way that the bigger the value of k is chosen, the smaller the value of h^2 becomes. Furthermore, we also used λ_{k+1} as the value of h^2 in the MQDF1 and evaluated the performance for various k instead of for various h^2. This is for comparing the performance of MQDF1 and MQDF2 as the same parameter is varied.

In Experiments I and II, 16- and 64-dimensional feature vectors were used, which were obtained from the Chinese character data in the ETL8 database (vol. 6 for test, vol. 7-32 for learning).[2] The feature vector is composed of the histograms of the direction code at the pixels on the contour of the partial patterns involved in the 2×2 (4×4) subsquares of a binary pattern (Fig. 1). More detailed description about the process is given in [10].

The number of classes was 927 for the 16-dimensional case and 271 for the 64-dimensional case (except for Experiment III). It was for reducing the computation cost that we decreased the number of classes for the 64-dimensional case. For each class, 130 characters were used for learning, and 5 characters per class for testing.

A. Experiment I

Table I shows the classification rates for the Euclidean distance, the QDF, and the MQDF2. The value of k and h^2 was set to 20 and λ_{21}, respectively, which were selected based on the result of Experiment II. For this result, we can see that the MQDF2 is less sensitive to the estimation error in the covariance matrix than the QDF. To see this point more precisely, we divided the 64-dimensional space into four subspaces ${}_kS_{k+15}$, each of which is spanned by 16 eigenvectors φ_i ($i = k, k+1, \cdots, k+15$), ($k = 1, 17, 33, 49$) and we evaluated the performance of QDF in each subspace. The QDF in each subspace is given as

$$g_0^k(x) = \sum_{i=k}^{k+15} \frac{1}{\lambda_i} \{\varphi_i^t(x - \mu_M)\}^2 + \log \prod_{i=k}^{k+15} \lambda_i$$
$$k = 1, 17, 33, 49. \tag{14}$$

For comparison, we also evaluated the performance of the Euclidean distance in each subspace, which was equivalent to (14) with all λ_i's equal to 1. Results are shown in Fig. 2(a), (b). From these results, we can see the following points.

1) For the learning samples, the QDF gives better performance in any subspace than the Euclidean distance.

2) For test samples, QDF gives poorer performance in subspace ${}_{33}S_{48}$ and ${}_{49}S_{64}$, although it gives better performance in the subspace ${}_1S_{16}$.

From these results, we can justify the MQDF2 and realize that the best performance is obtained by the MQDF2, which is equal to the QDF (Euclidean distance) in the subspace ${}_1S_k$ (${}_{k+1}S_n$), respectively.

Because the MQDF2 is an approximation of the MQDF1 and both employ almost the same eigenvalues, the MQDF1 can be also justified by the above results.

[2]One volume involves five data sets, each of which consists of 956 classes of characters. Among the classes, 881 elementary Chinese characters (KANJI) and 46 Hiraganas were used for experiment. The detailed description and specifications of the ETL8 database are presented in [9].

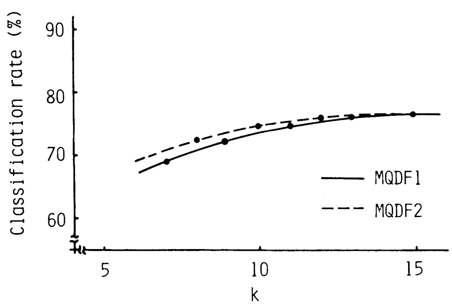

Fig. 3. Classification rate for MQDF's for various values of k (16-dimensional case, ETL8 database, vol. 6).

TABLE II
CLASSIFICATION RATES (PERCENT) (n = 16)

Euclidean distance	City-block distance	LDF*	QDF	MQDF1 (k=10)	MQDF2 (k=10)
46.4	46.2	52.5	76.4	73.7	74.8

* Linear discriminant function

B. Experiment II

Results for 16-dimensional case are shown in Fig. 3 and Table II. The solid line and the broken line in Fig. 3 show the classification rate for the MQDF1 and the MQDF2, respectively.

Fig. 3 shows

1) that there is no distinct difference between the classification rates for the MQDF1 and the MQDF2, and

2) that for the MQDF2 with $k = 10$, the classification rate nearly equal to that for the QDF (equivalent to the MQDF2 with $k = 16$) is achieved with required computation time and storage which are $\frac{10}{16}$ times those by QDF.

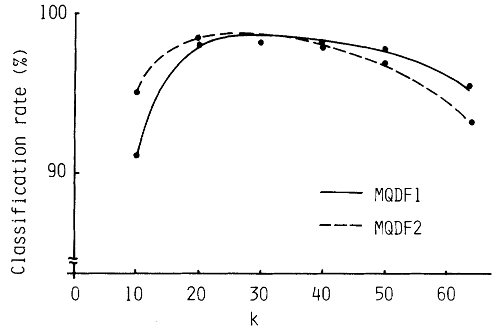

Fig. 4. Classification rate for MQDF's for various values of k (64-dimensional case, ETL8 database, vol. 6).

TABLE III
CLASSIFICATION RATES (PERCENT) ($n = 64$)

Euclidean distance	City-block distance	LDF	QDF	MQDF1 (k=30)	MQDF2 (k=20)
85.2	87.9	91.3	93.1	98.5	98.5

TABLE IV
CLASSIFICATION RATES (PERCENT) OF MQDF2 ($k = 20$)

Vol.1	Vol.3	Vol.5	Vol.6
99.3	99.5	98.6	97.1

TABLE V
AVERAGE COMPUTATION TIME (ms)

feature extraction	discrimination	total
66.0	668.5	734.5

Table II shows the classification rates by various discriminant functions for comparison. The rates by the MQDF1 and the MQDF2 are those for $k = 10$.

Fig. 4 and Table III show the results for the 64-dimensional case. In this case, the MQDF2 shows remarkable improvement in reducing the computation time and storage to the amount of $\frac{20}{64}$ and the error ratio rate of $\frac{1}{5}$. In Fig. 4, the classification rate begins to decrease for $k > 20$–30 because of the estimation error of the covariance matrix. Note that this is not exactly the peaking phenomenon which has been studied [1], [2], [5], [11] because the parameter k is not the dimensionality, but the number of the eigenvalue, the value of which is used as the value of h^2. The dimensionality is still n even if k is set to be a smaller value. The peaking phenomenon of MQDF1 or MQDF2 will be studied by changing the value of n. It will at least be not as severe as that of the QDF.

C. Experiment III

We carried out Experiment III to test the feasibility of the MQDF2 in a Chinese character OCR. In this experiment, 64-dimensional feature vectors were used to classify 20 characters per class into 927 classes. Table IV shows the classification rate for each database volume. This result is one of the best results obtained so far by Japanese researchers [9]. The variation of the rates for the volumes is due to the different quality of characters in each volume. For the results reported by several researchers [12], volume 1 can be regarded as one of the best representative of high-quality characters in the ETL8, and volume 6 as that of low-quality characters.

Table V shows the average computation time required by general-purpose digital computer FACOM M200 to read one character. Since the major part of the calculation in the MQDF2 consists of inner products, it is easy to achieve higher recognition speed by designing special-purpose hardware.

IV. CONCLUSION

The following was shown by experiments.

1) For learning samples, the QDF gives better performance in any subspace than the Euclidean distance.

2) But for the test samples, the QDF gives poorer performance in subspaces spanned by nondominant eigenvectors than the Euclidean distance because, in these subspaces, the QDF is more sensitive to the estimation errors than is the Euclidean distance.

3) MQDF's (MQDF1, MQDF2) are less sensitive to the estimation error in the covariance matrix than the QDF employing the maximum likelihood estimate and can achieve better performance.

Results 1) and 2) coincide with the theoretical consideration of the estimation errors of the eigenvectors as described in Section II. They are important to explain the behavior of the MQDF2 and to attempt further improvement of the QDF.

Other features of MQDF's are as follows.

4) They are reasonably used, even in the case that a sufficient number of samples is not available (and even in the case where the covariance matrix is singular),

5) The computation time and storage required by the MQDF2 are considerably smaller than those of QDF (about k/n times),

6) The MQDF2 is suitable for implementation by hardware.

The following points remain to be studied in the future.

1) The peaking phenomenon of the MQDF's for various feature sizes.

2) The choice of h^2 and k.

So far, we have used $\bar{\lambda}_{k+1}$ as the value of h^2 and have chosen the value of k experimentally. It is desirable to be chosen theoretically. For example, the value of k should be chosen so that the estimation error of eigenvectors φ_i ($i = 1, 2, \cdots, k$) does not become too large.

3) Further theoretical and experimental consideration.

Because our purpose is the application to Chinese character recognition and a Chinese character has a larger number of classes, we used only five test samples per class in Experiments I and II. We are going to deal with two-class problems to study theoretically and to conduct more extensive experiments by Monte Carlo data simulation.

ACKNOWLEDGMENT

We would like to sincerely thank Prof. J. Toriwaki, M. Yoshimura, and Associate Professor S. Yokoi for their helpful comments.

REFERENCES

[1] J. M. Van Campenhout, "On the peaking of Hughes mean recognition accuracy: The resolution of an apparent paradox," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-8, pp.390–395, May 1978.

[2] W. G. Waller and A. K. Jain, "On the monotonicity of the performance of Bayesian classifiers," *IEEE Trans. Inform. Theory*, vol. IT-24, pp. 392–394, 1978.

[3] K. A. Brakke, J. M. Mantock, and K. Fukunaga, "Systematic feature extraction," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-4, pp. 291–297, May 1982.

[4] R. O. Duda and P. E. Hart, *Pattern Classification and Scence Analysis.* New York: Wesley, 1973, p. 67.

[5] R. P. W. Duin and B. J. Kröse, "On the possibility of avoiding peaking," in *Proc. 5th Int. Conf. Pattern Recognition*, Miami, FL, 1980, pp. 1375–1378.

[6] J. V. Ness, "On the dominance of non-parametric Bayes rule discriminant algorithms in high dimensions," in *Pattern Recognition, Vol. 12.* New York: Pergamon, 1980, pp. 355–368.

[7] K. Fukunaga, *Introduction to Statistical Pattern Recognition.* New York and London: Academic, 1972.

[8] A. G. Wacker and T. S. El-Sheikh, "Average classification accuracy

over collections of Gaussian problems: Common covariance matrix case," in *Pattern Recognition, Vol. 17.* New York: Pergamon, 1984, pp. 259-273.
[9] S. Mori, K. Yamamoto, and M. Yasuda, "Research on machine recognition of handprinted characters," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, pp. 386-405, July 1984.
[10] M. Kurita *et al.*, "Handprinted KANJI and HIRAGANA character recognition using weighted direction index histograms and quasi-Mahalanobis distance," *Trans. IECE Japan*, vol. PRL82-79, Jan. 1983.
[11] G. F. Hughes, "On the mean accuracy of statistical pattern recognizers," *IEEE Trans. Inform. Theory*, vol. IT-14, pp. 55-63, Jan. 1968.
[12] T. Saito, H. Yamada, and K. Yamamoto, "An analysis of handprinted Chinese character by directional pattern matching approach," *J. IECE Japan*, vol. J65-D, pp. 550-557, 1982.

Semantic Network Array Processor and Its Applications to Image Understanding

V. DIXIT AND D. I. MOLDOVAN

***Abstract*—The problems in computer vision range from edge detection and segmentation at the lowest level to the problem of cognition at the highest level. This correspondence describes the organization and operation of a semantic network array processor (SNAP) as applicable to high level computer vision problems. The architecture consists of an array of identical cells each containing a content addressable memory, microprogram control, and a communication unit. The applications discussed in this correspondence are the two general techniques, discrete relaxation and dynamic programming. While the discrete relaxation is discussed with reference to scene labeling and edge interpretation, the dynamic programming is tuned for stereo.**

***Index Terms*—Discrete relaxation, labeling, parallel processing, semantic networks, symbolic processing.**

I. Introduction

The nature of symbolic processing used in artificial intelligence (AI) is different in many ways from conventional programming language processing. Consequently, the architecture of computers intended for AI applications should be different from today's commonly used von Neumann computers. The mapping of AI algorithms into architectures cannot be done with the same efficiency as that of numerical signal processing algorithms (mapping into systolic arrays, for example). Communication networks supporting packet switching and complicated data transfer protocols are necessary.

The vision algorithms range from very low number crunching to symbolic processing. It is not possible to efficiently implement all these algorithms on a single machine. SNAP (*S*emantic *N*etwork *A*rray *P*rocessor) currently under study at USC, addresses the high end of the vision processing. In this correspondence we show how SNAP can be effectively used for discrete relaxation and dynamic programming for stereo. The interested reader should see [9] for other symbolic processing applications such as pattern search, inference, and production systems.

Manuscript received January 21, 1985; revised May 8, 1986. Recommended for acceptance by R. Bajcsy. This work was supported by the Defense Advanced Research Projects Agency under Contracts F-33615-82-K-1786 and F-33615-84-K-1404.
The authors are with Department of Electrical Engineering—Systems, University of Southern California, Los Angeles, CA 90089.
IEEE Log Number 8609968.

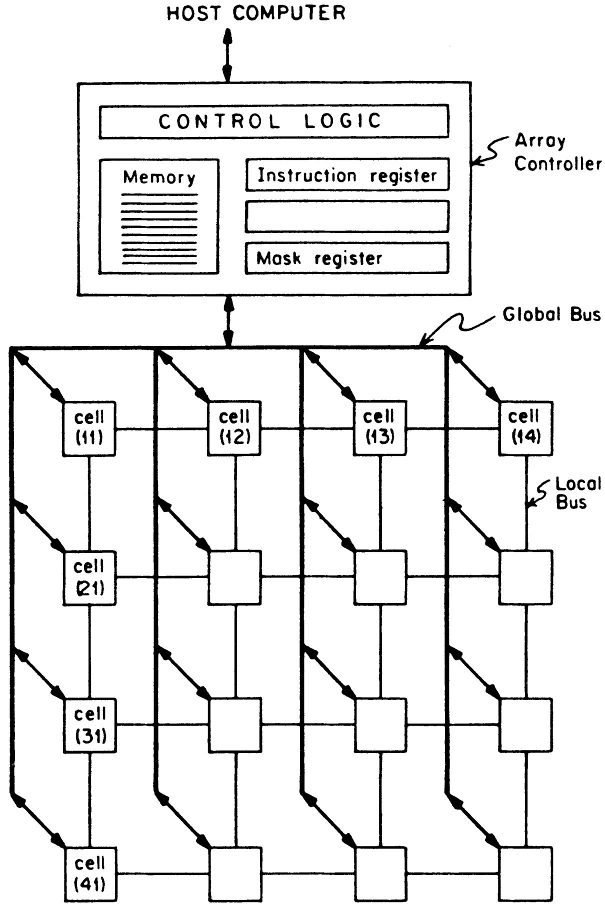

Fig. 1. Architecture of SNAP.

In this correspondence we first present briefly the architecture and the instruction set of SNAP, and then show how discrete relaxation and dynamic programming can be processed on SNAP.

II. SNAP Architecture

The architecture consists of a square array of identical processing cells which are interconnected both globally and locally as shown in Fig. 1.

Its functionality rests upon two underlying concepts: associative processing and cellular array processing. Each cell contains memory control logic and communication logic. As a whole, the array is operated by an outside controller which also provides an interface between SNAP and a host computer. Our intent was to minimize the role of the global functions which affect the entire array and to provide more operational freedom for each individual cell. The cells can be microprogrammed so they can operate independently. The signals involved in the intercell communications are propagated from a cell to another neighboring cell via local buses. As a result, any cell can communicate through a chain of intermediate cells to any other cell in the array. A cell's address is specified by its row number followed by its column number. Information in a particular cell can be retrieved either by its content (as in associative memories) or by that cell's address. The associative processing concept alone is not sufficient for designing efficient architectural structures for AI. This is because retrieval operations in AI are more complex than simple words; most freqeuntly we need to match subgraphs or other patterns. Also, we need to pursue several hypotheses in parallel, and this is not possible with simple associative processors. This architecture blends the power of associative memories for performing fast information retrievals with the capability of cellular array to process various tasks in parallel.

0162-8828/87/0100-0153$01.00 © 1987 IEEE