

- [14] J. E. Kaufman, *IES Lighting Handbook*, 4th ed. New York: Illuminating Eng. Soc., 1966.
- [15] S. Kawata, K. Sasaki, and S. Minami, "Component analysis of spatial and spectral patterns in multispectral images. I. Basis," *J. Opt. Soc. Am. A*, vol. 4, pp. 2101-2106, 1987.
- [16] G. J. Klinker, S. A. Shafer, and T. Kanade. "The measurement of highlights in color images," *Int. J. Comput. Vision*, vol. 2, pp. 7-32, 1988.
- [17] W. H. Lawton and E. A. Sylvestre, "Self modeling curve resolution," *Technometrics*, vol. 13, pp. 617-633, 1971.
- [18] H.-C. Lee, "Method for computing the scene-illuminant chromaticity from specular highlights," *J. Opt. Soc. Am. A*, vol. 3, pp. 1694-1699, 1986.
- [19] S. K. Nayar and Y. Gong, "Colored interreflections and shape recovery," in *Image Understanding Workshop*, in DARPA, San Diego, CA 1992.
- [20] S. K. Nayar, K. Ikeuchi, and T. Kanade, "Shape from interreflections," in *Proc. IEEE Int. Conf. Comput. Vision*, Osaka, Japan, Dec.4-7, 1990, pp. 2-11.
- [21] S. A. Shafer, "Using color to separate reflection components," *Color Res. Appl.*, vol. 10, pp. 210-218, 1985.
- [22] S. A. Shafer, T. Kanade, G. J. Klinker, and C. L. Novak, "Physics-based models for early vision by machine," in *Perceiving, Measuring, and Using Color*. New York: SPIE, 1990, vol. 1250, pp. 222-235.
- [23] R. Siegel and J. R. Howell, *Thermal Radiation Heat Transfer*. New York: Hemisphere, 1981.
- [24] S. Tominaga and B. A. Wandell, "Standard surface-reflectance model and illuminant estimation," *J. Opt. Soc. Am. A*, vol. 6, pp. 576-584, 1989.
- [25] —, "Component estimation of surface spectral reflectance," *J. Opt. Soc. Am. A*, vol. 7, pp. 312-317, 1990.
- [26] F. Tong and B. V. Funt, "Specularity removal for shape from shading," in *Proc. Vision Interface 1988*, Edmonton, AB, Canada, 1988, pp. 98-103.
- [27] —, "Removing specularities from color images for shape from shading," in *Computer Vision and Shape Recognition*, A. Krzyzak, Ed. New York: World Scientific, 1989.
- [28] G. Wyszecki and W. S. Stiles, *Color Science: Concepts and Methods, Quantitative Data and Formulas*, 2nd ed. New York: Wiley, 1982.

### Vector Quantization Technique for Nonparametric Classifier Design

Qiaobing Xie, Charles A. Laszlo, and Rabab K. Ward

**Abstract**—An effective data reduction technique based on vector quantization is introduced for nonparametric classifier design. Two new nonparametric classifiers are developed, and their performance is evaluated using various examples. The new methods maintain a classification accuracy that is competitive with that of classical methods but, at the same time, yields very high data reduction rates.

**Index Terms**—Condensing algorithms, data reduction,  $k$ -nearest-neighbor ( $k$ NN) classifier, nonparametric classification, Parzen kernel classifier, vector quantization.

#### I. INTRODUCTION

Nonparametric classification has been of great importance in statistical pattern recognition [1], [2]. When dealing with problems

Manuscript received September 23, 1991; revised June 2, 1992. Recommended for acceptance by Editor-in-Chief A. K. Jain.

The authors are with the Department of Electrical Engineering, University of British Columbia, Vancouver, Canada V6T 1Z4.

IEEE Log Number 9211830.

of nonnormal distributions, nonparametric classifiers frequently show much higher classification accuracy than that achieved by the popularly used parametric classification approaches, such as the linear classifier and the quadratic classifier. In addition, it has been found that the nonparametric algorithms sometimes outperform their parametric counterparts even when the data are from normal populations [3], [4]. In practice, however, the application of the nonparametric classifiers often suffers from various difficulties, especially as the size of the problem increases.

The common disadvantages of classical nonparametric approaches (kernel estimator,  $k$ NN classifier, etc.) are their computational complexity and the requirement for a very large amount of computer storage to retain the design sets. Unfortunately, large design sets are always desirable because nonparametric methods require adequate statistical information. Thus, on-line application of nonparametric classifiers is rare since these are usually too complex and slow in operation. Their uses are often limited to situations where the computation time is not a crucial factor, such as in the estimation of the Bayes error and data structure analysis [1].

A solution to the above problems is to reduce the size of the design set while insisting that the classifiers built on the reduced design set should perform as well, or nearly as well, as the classifiers built on the original design set. This idea has been explored for various purposes over a period of time and resulted in the development of many algorithms of  $k$ NN classifier design using reduced sample sets. Particularly noteworthy are the condensed NN (CNN) [5], the reduced NN (RNN) [6], and the edited NN (ENN) [7]. In these algorithms, iterative processes are used to test the effect on the classification performance as each individual point is moved in and out of the design set, and only the "good" points are retained. For very large design sets, these methods are often tedious and difficult to implement since a new classifier is in fact built and evaluated every time a point is moved in or out of the design set. The most serious disadvantage is that the final reduction rate is usually low and not under the control of the algorithms, e.g., it depends entirely on the nature of the sample set to be reduced.

Recently, two nonparametric data reduction algorithms were proposed by Fukunaga *et al.* for the Parzen's kernel classifier and the NN classifier design, respectively [1], [8], [9]. Their algorithms find the optimal reduced design set from the original design set in the sense that the difference between the probability density function estimated from the reduced set and that estimated from the original set is minimized. Bearing some similarities to the traditional reduced data  $k$ NN algorithms, their algorithms iteratively move each individual point in and out of a tentatively chosen reduced sample set and test the resultant effect on the criterion function. To avoid an exhaustive search of all possible subsets, which is impractical, the optimization scheme used in Fukunaga's algorithms can achieve a local optimum. The computational complexities of these algorithms are considerable. Moreover, the initial guess of the reduced sample set is of crucial importance in Fukunaga's reduced NN algorithm. Thus far, only an intuitively developed initial assignment procedure for the 2:1 reduction rate case has been published [9].

#### II. TWO NEW NONPARAMETRIC CLASSIFIERS USING VECTOR QUANTIZATION TECHNIQUE

In this article, we introduce a new approach for nonparametric data reduction using the vector, or block, quantization technique. As a mathematical process, optimal vector quantization has already been

widely used in various areas of engineering, such as digital signal processing, communication, and speech recognition. Our research indicates that this technique can also be effectively utilized to perform data reduction in the design of nonparametric classifiers. Combining vector quantization with the classical Parzen's Kernel and the  $k$ NN approaches, we develop two new algorithms of reduced nonparametric classifier design, which we shall denote the VQ-kernel and the VQ- $k$ NN methods.

In these algorithms, we first construct an optimal vector quantizer for each class of the training data. For each class, the corresponding original design set is used as the training sequence of the quantizer. Then, we utilize the resulting reproduction alphabets, or code-books, of the quantizers as the reduced design sets to represent the original sets. With the reduced sets, a classifier using either the Parzen's kernel or the  $k$ NN methods is then built.

For our purposes, we find that the reproduction alphabet serves as a good representative of the original design set. The obtainable reduction rate can be significantly high and may be preset freely in the algorithms. As illustrated by the examples given later, large original design sets containing hundreds of vectors can be well represented by a reproduction alphabet with only a few vectors. Despite the high reduction rate, our VQ-based classifiers perform as well or better than classical reduction methods in terms of the classification error rate. The remarkable performance for the high data reduction rate probably results from the fact that our method does not restrict the search for the members of the representative (reduced) design set to points in the original design set. Instead, our method generates the representative set using the information from all the points in the original set.

In our VQ-based methods, we use the algorithm developed by Linde *et al.* [10] for vector quantizer design. This algorithm has the advantage that it is well developed and widely applied in various field, and its efficiency in implementation and its convergence properties are proven.

As an alternative, the *learning vector quantization* (LVQ) algorithm, which is proposed recently by Kohonen [11], [12], may be used to generate the quantizer. The overall performance of the LVQ technique has been shown to be comparative with that of Linde's VQ algorithm for applications such as image compression [13].

### III. DEVELOPMENT OF THE ALGORITHMS

Since our two algorithms use vector quantization as the first stage, it is helpful to begin our presentation with an introduction to the vector quantization theory. Then, in Section III-B, we will discuss a relevant property of vector quantization. Our algorithms are then described in Section III-C.

#### A. Some Definitions of Vector Quantization

An  $M$ -level  $d$ -dimension quantizer is a mapping  $q(\bullet)$  that assigns to each input vector  $\mathbf{x} = (x_0, \dots, x_{d-1})$ , a reproduction vector,  $\mathbf{y}_k = q(\mathbf{x})$ , that is drawn from a finite reproduction alphabet (or codebook)  $\hat{A} = \{\mathbf{y}_i; i = 1, \dots, M\}$ . The  $q(\bullet)$  is usually chosen as a *minimum-distance mapping*. This means that the reproduction vector  $\mathbf{y}_k$  is chosen such that

$$d(\mathbf{x}, \mathbf{y}_k) = \min_{\text{all } i} d(\mathbf{x}, \mathbf{y}_i) \quad (1)$$

where  $d(\bullet, \bullet)$  is any nonnegative distance measure defined on the  $d$ -dimension space, and  $d(\mathbf{x}, \mathbf{y}_k)$  is called the distortion of the quantization. Obviously, a minimum-distance mapping, which is completely defined by the reproduction alphabet  $\hat{A}$  along with the definition of the distance measure, uniquely describes a Dirichlet partition  $S = \{S_i; i = 1, \dots, M\}$ , of the sample space.

An  $M$ -level quantizer is said to be optimal if it minimizes the expected distortion  $D(q) = E\{d(\mathbf{x}, q(\mathbf{x}))\}$ , that is,  $q^*$  is optimal if for any other quantizer  $q$  having  $M$  reproduction vectors,  $D(q^*) \leq D(q)$ .

#### B. Distribution Property of the Reproduction Vectors and the VQ Data Reduction Method

In the following, we argue that the distribution of the reproduction vectors in an optimal vector quantizer possesses desirable properties that make the VQ technique a promising approach of nonparametric data reduction. Before doing so, however, it is important to point out a common misunderstanding about the nonparametric classifier design.

The philosophy guiding the development of most traditional nonparametric classification methods is that of using the statistical information contained in a set of preclassified samples (or design set) for finding a good approximation of the actual underlying probability density function  $p(\mathbf{x})$ ; then, the classifier is built by applying the Bayesian rule. This philosophy is also explicitly employed in the development of Fukunaga's reduced Parzen and NN classifiers, in which, as mentioned earlier, the reduced design set is selected in such a way that the difference between the density estimate obtained from this reduced set and that obtained from the original design set is minimized. However, for achieving high classification performance, *this approximation to  $p(\mathbf{x})$ , although it is obviously sufficient, is not necessary*. An example is that any good approximation to  $[p(\mathbf{x})]^\alpha$ , where constant  $\alpha > 0$ , will achieve the same Bayesian classifier as that achieved by approximating  $p(\mathbf{x})$  itself.

In [14], Gersho addresses the properties of the reproduction vectors of an optimal vector quantizer. The density function of the reproduction vector in a  $d$ -dimensional quantizer is defined as

$$g_M(\mathbf{x}) = \frac{1}{MV(S_i)}, \quad \text{if } \mathbf{x} \in S_i, \quad \text{for } i = 1, 2, \dots, M, \quad (2)$$

where  $V(S_i)$  is the volume of  $S_i$ . Gersho shows that *for an optimal quantizer, in the asymptotic situation where  $M$  is sufficiently large,  $g_M(\mathbf{x})$  will closely approximate a continuous density function  $\lambda(\mathbf{x})$ , which is proportional to  $[p(\mathbf{x})]^{1+\beta}$ , where  $p(\mathbf{x})$  is the actual underlying density function of the input random vector, and  $\beta$  is a constant determined by the dimension  $d$  and the distance measure*.

This finding, along with our argument at the beginning of this section, strongly indicates that the reproduction alphabet in an optimal quantizer could be used as an effective design set for building classifiers, provided that the level of the quantizer is sufficiently high.

With VQ algorithms such as Linde's and Kohonen's, we can estimate an optimal (either locally or globally) reproduction alphabet from a set of training data. This leads us to the following data reduction approach for the nonparametric classifier design: Generate the reduced set as the optimal reproduction alphabet found by using an optimal VQ method using the original design set as the training data. In the following section, we describe our data reduction methods in full detail.

The most important difference between our data-reduction method and the other traditional methods is that our reduced set is *not* necessarily a subset of the original design set. With our method, the vectors in the reduced set are *created (not selected)* by surveying all the information carried by every known vector. These vectors are created so that they best represent all the vectors in the original set in the sense of minimizing the average distortion rate of quantization.

#### C. Classifier Design Algorithms Using Vector Quantization

*The VQ-kernel Classifier:* In this method, we propose that vector quantization be first applied to the original design set of each class. The reproduction alphabets of the resultant optimal quantizers are

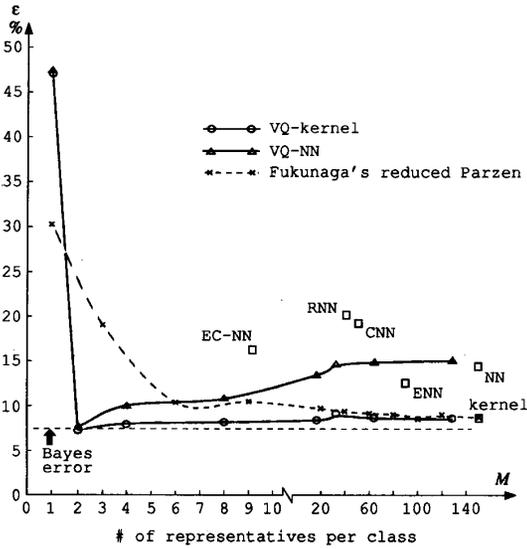


Fig. 1. Classification error rates of our VQ-based classifiers and other traditional reduced data classifiers.

then retained as the reduced design sets. Then the kernel method is applied as usual except that the reduced design sets are used.

The following steps explain this method in greater detail.

Given an  $N$ -class problem in  $d$  dimension, for each class  $w_i$  ( $i = 1, \dots, N$ ), an original design set  $\{\mathbf{x}_j^{(i)}; j = 0, \dots, n_i - 1\}$  is assumed given.

- 1) For class  $w_i$ , as in the algorithm in [10], we start with an initial  $M_i$ -level reproduction alphabet  $\hat{A}_{0i}$ . This initial alphabet can be generated by using the "splitting" approach proposed in [10].
- 2) Find the  $M_i$ -level optimal quantizer for class  $w_i$  using the quantizer design method in [10]. Suppose the final reproduction alphabet for class  $w_i$  is  $\hat{A}_i = \{\mathbf{y}_j^{(i)}; j = 1, \dots, M_i\}$ .
- 3) Apply the Parzen's kernel method to the reproduction alphabet  $\hat{A}_i$  to get an estimate of  $p(\mathbf{x} | w_i)$ , i.e.,

$$\hat{p}(\mathbf{x} | w_i) = \frac{1}{M_i} \sum_{j=1}^{M_i} K(\mathbf{x} - \mathbf{y}_j^{(i)}), \quad (3)$$

where  $K$  represents the kernel function.

- 4) Repeat steps 1 to 3 for each class and get estimates of  $\hat{p}(\mathbf{x} | w_i)$  for  $i = 1, \dots, N$ .
- 5) Finally, a Bayes classifier is built on all the estimated class-conditional pdfs ( $\hat{p}(\mathbf{x} | w_i)$ ). The Bayes classifier assigns the unknown observation  $\mathbf{x}$  to class  $w_m$  if  $p(\mathbf{x} | w_m)P(w_m) \geq p(\mathbf{x} | w_l)P(w_l)$  for all  $l \neq m$ , where  $P(w_i)$  is the known *a priori* probability of class  $w_i$ .

For the special case when the levels of quantizers are chosen equal to the number of vectors in the original design sets, i.e.,  $M_i = n_i$  ( $i = 1, \dots, N$ ), the above VQ-kernel method becomes equivalent to the traditional kernel method.

It should be noted that the selection of the kernel function as well as its parameters is important yet difficult. This disadvantage is inherent to the classical kernel approach. Theories and conclusions developed in the literature on the classical kernel methods can be directly used to guide the selection of proper kernels and smoothing factors. A thorough discussion on this topic is found in [1] and [15].

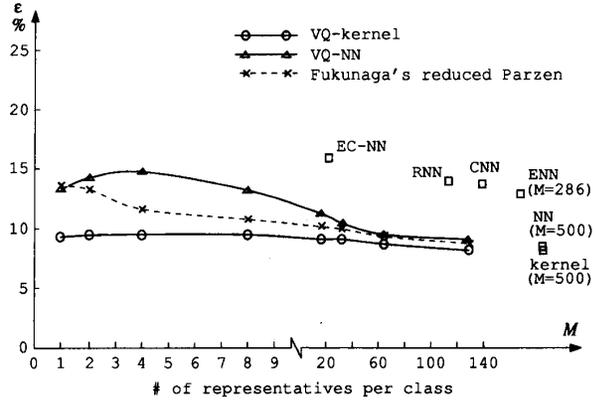


Fig. 2. Classification error rates of our VQ-based classifiers, the traditional reduced data NN classifiers, and Fukunaga's reduced Parzen classifier for the speech data.

**The VQ-kNN Classifier:** This algorithm combines the VQ technique and the  $k$ NN method.

For an  $N$ -class problem, we assume that for each class  $w_i$  ( $i = 1, \dots, N$ ), a design set  $\{\mathbf{x}_j^{(i)}; j = 0, \dots, n_i - 1\}$  and an initial  $M_i$ -level reproduction alphabet  $\hat{A}_{0i}$  are given (for the selection of  $\hat{A}_{0i}$  see [10]).

- 1) For each class  $w_i$  ( $i = 1, \dots, N$ ), find the  $M_i$ -level optimal quantizer using the algorithm in [10]. Suppose the final reproduction alphabets are  $\hat{A}_i = \{\mathbf{y}_j^{(i)}; j = 1, \dots, M_i\}$  ( $i = 1, \dots, N$ ).
- 2) Combine all the reproduction alphabets  $\hat{A}_i$ 's of all classes into one single set  $\hat{A}$  of  $M$  vectors, that is,  $\hat{A} = \bigcup_i \hat{A}_i$  and  $M = \sum_i M_i$ .
- 3) The classification rule is as follows: Assume  $\mathbf{x}$  is the new observation point to be classified.

- a) For  $\mathbf{x}$ , find the first  $k$  nearest reproduction vectors in  $\hat{A}$ . Suppose that amongst these  $k$  reproduction vectors there are  $k_m$  vectors from class  $w_m$ . Then, the classification rule is assign  $\mathbf{x}$  to class  $w_m$  if

$$P(w_m) \frac{k_m}{M_m} \geq P(w_l) \frac{k_l}{M_l} \quad \text{for all } l \neq m. \quad (4)$$

- b) If the *a priori* probability  $P(w_i)$  is unknown, we can approximate it from the design set as

$$\hat{P}(w_i) = \frac{n_i}{\sum_{i=1}^N n_i}. \quad (5)$$

The classification rule, then, becomes the following: Assign  $\mathbf{x}$  to class  $w_m$  if

$$\frac{n_m k_m}{M_m} \geq \frac{n_l k_l}{M_l} \quad \text{for all } l \neq m. \quad (6)$$

The classical  $k$ NN method becomes a special case of our new method when  $M_i$  is chosen to be equal to  $n_i$  ( $i = 1, \dots, N$ ). If  $k = 1$  is used, we get the VQ-NN classifier, which is very effective in computation since when all the *a priori* probabilities are equal, the unknown observation is simply assigned to the class to which its nearest reproduction vector belongs.

TABLE I  
CPU TIME USED FOR FINDING THE REDUCED SET IN THE SPEECH DATA EXPERIMENT

Classifier	VQ-NN	CNN	ENN	ECNN	RNN	VQ-kernel*	Fukunaga's kernel*
CPU Time (min)	3.41	5.31	21.17	22.17	51.34	3.92	3378.45

\*including CPU time used for classification.

#### IV. EXAMPLES

We give two examples to demonstrate the use of our VQ-kernel and VQ-NN classifiers. The data used in the first example has a known probability distribution that is a mixture of Gaussian distributions. In the other example, we test our VQ classifiers with real speech data of unknown probability distribution. For the reason of comparison, we also test the performances of those traditional reduction algorithms including the CNN, RNN, and ENN as well as Fukunaga's reduced Parzen.

*Example 1:* We adopt Fukunaga's 8-D data model from Experiment 11-7 on page 555 of [1]. In this model, each of the two classes consists of two Gaussian distributions. The *a priori* probabilities are equal for the two classes. The class-conditional pdfs are  $p(\mathbf{x} | \omega_1) = 0.5N(\mu_1, I) + 0.5N(\mu_2, I)$  and  $p(\mathbf{x} | \omega_2) = 0.5N(\mu_3, I) + 0.5N(\mu_4, I)$ , respectively, where the covariance matrices are equal to the identity matrix  $I$ , and the mean vectors are  $\mu_1 = [0 \ 0 \ \dots \ 0]^T$ ,  $\mu_2 = [6.58 \ 0 \ \dots \ 0]^T$ ,  $\mu_3 = [3.29 \ 0 \ \dots \ 0]^T$ , and  $\mu_4 = [9.87 \ 0 \ \dots \ 0]^T$ , respectively. The Bayes error of this data model is  $\varepsilon = 7.5\%$ . The Euclidean metric system was used as the distance measure. The original design set contains 150 vectors for each class, and similarly, the testing set has 150 vectors from each class.

Fig. 1 shows the result of our experiment in which we tested our VQ-kernel and VQ-NN classifiers as well as other traditional reduced data classifiers, including the CNN [5], the RNN [6], and the ENN and the edited-condensed nearest neighbor (EC-NN) [7]. Fukunaga *et al.* reported the performance of their reduced Parzen classifier with the same data model in [1]. We include their results in the graph as well. The same normal kernel function, with a constant covariance matrix of  $1.5^2 \times I$ , which was used in Fukunaga's reduced Parzen classifiers, was also used in our VQ-kernel method. Each point in the graph represents the average result after 10 trials.

In both Fukunaga's and our methods, all 1-level reduced data classifiers failed due to the fact that the underlying probability density of each class is composed of two separate Gaussian distributions. When two or more representatives are used in the reduced sets, both our new VQ-based classifiers showed excellent performance. In particular, our VQ-kernel classifier achieved the best performance amongst all the classifiers. It gave a classification accuracy extremely close to the Bayes error, which is the theoretical lowest bound, at almost all the reduction rates, and its performance showed little correlation to the reduction rate.

Our VQ-NN classifier evidently also outperformed all other modified NN classifiers in terms of both the reduction rate and the classification accuracy. This is with the exception of the ENN, which showed a better accuracy but at a low reduction rate of only 1.67:1. It is very interesting to notice that at high reduction rates ( $M < 40$ ), our VQ-NN classifier showed significantly better performance than it did at the lower reduction rates. We believe that this is due to the specific structure of the distribution density functions (mixtures of two Gaussian distributions) underlying the experimental data.

*Example 2:* This example demonstrates the classification performance of our new VQ-based classifiers with real data extracted from speech signals.

The 12-D speech data measured the first 10 cepstrum coefficients, the short-time zero-crossing rate, and the short-time energy of the voice signals.

The data for class 1 (2247 vectors in total) came from male speaker 1 (*Chuck*), and the data in class 2 (2256 vectors in total) were for male speaker 2 (*Gray*). For each trial in the experiments, we first randomly drew a 500-vector design set from each class and used the rest to form the testing set.

For our VQ-kernel classifier and Fukunaga's reduced Parzen classifier, we used the Euclidean metric system and normal kernel functions with their covariance matrices estimated from the original design sets, i.e.,  $\sum_i = C_i$  for  $i = 1, 2$ , where  $C_i$  is the covariance matrix of the original design set of class  $i$ . For our VQ-NN and the traditional reduced data NN classifiers, the variance-weighted distance measure was used, i.e.,

$$d(\mathbf{x}, \mathbf{y}) = \sum_l |x_l - y_l|^2 / \sigma_l, \quad (7)$$

where  $x_l$  and  $y_l$  are the  $l$ th element of the  $x$  and  $y$ , respectively, and  $\sigma_l$  is the variance of the  $l$ th element of the training vectors.

Fig. 2 shows the result of our experiments. The best performances are given by our VQ-kernel classifier at all the reduction rates. Again, little dependency between the classification accuracy of our VQ-kernel classifier and the reduction rate was shown. Compared with Fukunaga's reduced Parzen classifier, the improvement on the classification accuracy of our VQ-kernel classifier is obvious, especially at the high reduction rate ( $> 25:1$  or  $M < 20$ ).

When operating at the same range of reduction rate (for  $M \geq 16$ ), our VQ-NN classifier significantly outperformed the other reduced data NN classifiers. In addition, for  $M \geq 32$ , our VQ-NN classifier gave statistically identical classification accuracy as Fukunaga's reduced Parzen classifier. At much higher reduction rates (62.5:1 to 500:1, i.e.,  $1 \leq M \leq 8$ ), the classification accuracy of our VQ-NN classifier was still comparable with those of the other reduced data NN classifiers at much lower reduction rates, ranging from 1.75:1 for the ENN to 23:1 for the EC-NN.

In addition to the classification accuracy and the reduction rate, the computational intensity of Fukunaga's algorithms is tremendously larger than that of our VQ-kernel algorithm. The computational complexity of finding the reduced set in Fukunaga's reduced Parzen algorithm can be shown to be of the order of  $rN^2k^2$ , whereas that of our VQ-kernel algorithm is of the order of  $rNk$ , where  $r$  is the size of the reduced set,  $N$  is the size of the original set, and  $k$  is the dimension of the data. In our above speech data experiment, a single trial of our VQ-kernel classifier took only about 3.92 min of CPU time, whereas a trial of Fukunaga's reduced Parzen classifier for exactly the same data and on the same SUN Sparc 2 computer took over 56 hr of CPU time! Amongst the NN-based algorithms, our VQ-NN classifier was also found to be the fastest. Table I shows the CPU time used in a single trial by each of the tested algorithms

for constructing the reduced design sets in the above speech data experiment. The CPU time for our VQ-NN was measured with  $M = 128$ .

## V. CONCLUSIONS

We introduce the vector quantization technique into the area of nonparametric classifier design and show that vector quantization is an extremely effective approach to data reduction.

Using the vector quantization data reduction technique, two methods of nonparametric classifier design, namely, the VQ-kernel and the VQ-k-NN, are proposed and tested with both synthetic and real data with various attributes. Compared with other known nonparametric data reduction algorithms, the new methods are found:

- 1) to give much better results in terms of both the classification accuracy and the data reduction rate;
- 2) to have significantly less computational complexity in general;
- 3) to have control over the reduction rate;
- 4) to achieve a classification accuracy which is only moderately dependent on the reduction rate.

Theoretically, for highly nonparametric data the classification accuracy of our VQ-based classifiers will increase as  $M_i$ , the number of representatives in the reduced set, is increased. When the value of  $M_i$  approaches that of the size of the original design set, the performance of our VQ-based classifiers will approach that of the basic Parzen and NN classifiers. This tendency is shown clearly in our experiments above. Therefore, the selection of  $M_i$  is a tradeoff: A larger  $M_i$  generally yields higher accuracy but lower reduction rate, and a smaller  $M_i$  yields lower accuracy but higher reduction rate.

## REFERENCES

- [1] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed. New York: Academic, 1990.
- [2] J. Hand, *Discrimination and Classification*. New York: Wiley, 1981.
- [3] J. W. V. Ness, "On the effects of dimension in discriminant analysis for unequal covariance populations," *Technometrics*, vol. 21, pp. 119-127, 1979.
- [4] J. W. V. Ness and C. Simpson, "On the effects of dimension in discriminant analysis," *Technometrics*, vol. 18, pp. 175-187, 1976.
- [5] P. E. Hart, "The condensed nearest neighbor rule," *IEEE Trans. Inform. Theory*, vol. IT-14, pp. 515-516, 1968.
- [6] G. W. Gates, "The reduced nearest neighbor rule," *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 431-433, 1972.
- [7] P. A. Devijver and J. Kittler, "On the edited nearest neighbor rule," in *Proc. 5th Int. Conf. Pattern Recogn.*, 1980, pp. 72-80.
- [8] K. Fukunaga and R. R. Hayes, "The reduced Parzen classifier," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-11, pp. 423-425, Apr. 1989.
- [9] K. Fukunaga and J. M. Mantock, "Nonparametric data reduction," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, pp. 115-118, Jan. 1984.
- [10] Y. Linde, A. Buzo, and R. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COM-28, pp. 84-95, Jan. 1980.
- [11] T. Kohonen, "Learning vector quantization," *Neural Net.*, vol. 1, pp. 303, 1988, Supplement 1.
- [12] T. Kohonen, G. Barna, and R. Chrisley, "Statistical pattern recognition with neural networks: Benchmarking studies," in *IEEE Proc. ICNN'88*, 1988, pp. 61-68, vol. 1.
- [13] N. Nasrabadi and Y. Feng, "Vector quantization of images based upon the Kohonen self-organizing feature maps," in *IEEE Proc. ICNN'88*, 1988, pp. 1101-1108, vol. 2.
- [14] A. Gersho, "Asymptotically optimal block quantization," *IEEE Trans. Inform. Theory*, vol. IT-25, pp. 373-80, July 1979.
- [15] D. J. Hand, *Kernel Discriminant Analysis*. New York: Research Studies, 1982.

## Comments on "Design of Fiducials for Accurate Registration Using Machine Vision"

G. Chiorboli and G.P. Vecchi

**Abstract**—Subpixel registration accuracy improves by one order of magnitude if images are not binarized as indicated in the referenced paper, but gray scale information is fully exploited in calculating centroid position.

**Index Terms**—Machine vision, image processing, positional accuracy, subpixel registration, camera calibration.

Measurement of the position of fiducial landmarks with subpixel precision was recently discussed in [1], with reference to binary images obtained by straight thresholding of the gray level picture. Several shapes were compared in terms of maximum and r.m.s. error, and the advantages of the circular shape were pointed out.

In connection with some activity on camera calibration, we came across the same problem of accurately measuring the position of circular dots patterned on a reference plate (a chromium-quartz IC mask) and found an advantage of one order of magnitude in exploiting the gray level information available.

This may be easily understood with reference to the simple, one-dimensional case depicted in Fig. 1. A short bar — 1 pixel long, 100% intensity over a null background — is the object whose position has to be determined using the 2-pixel sensor represented at the left of Fig. 1(a). When the position of the bar is varied, as shown by the vertical segments at the right of the sensor, the signal provided by the two photosites varies in the range 0-1, in proportion of the fraction of the photosite covered by the bar. It can be demonstrated that the maximum absolute error in the position of the center of mass, expressed in pixel units, is  $\epsilon_{\max} = |s - 0.25| + 0.25$  in the case of straight binarization by thresholding at level  $s$ ,  $0 \leq s \leq 0.5$ , while  $\epsilon_{\max} = s$  if the center of mass is obtained by weighting the response of each photosite above threshold with the difference  $(y - s)$  between the photosite signal  $y$  and the threshold  $s$ . These results hold as long as  $s \leq 0.5$ , because the bar length was assumed equal to the one pixel, and are represented by the graphs of Fig. 1(b), which report the value of the error  $\epsilon$  in the estimated position of the center of mass as a function of the position  $x$  of the lower end of the bar, in the two cases of binarization and of gray-scale weighting. For thresholds greater than 0.5, the binarized bar would disappear in some positions, because of the threshold too high. The mean absolute value of error  $|\bar{\epsilon}|$  is  $|\bar{\epsilon}| = 2(s - 0.25)^2 + 0.125$  in the case of binarization, and  $|\bar{\epsilon}| = s^2$  in the gray scale case, see Fig. 1(c). Thus, the mean absolute error when gray scale information is used, whichever the threshold's value, is smaller or equal than in the case of straight binarization.

Looking at a more realistic case, Fig. 2(a) shows the intensity plot measured for one of the dots of our reference plate: as may be noticed, the black-white transition occurs over a range of several pixels, even with a well focused image.

Manuscript received July 11, 1991; revised February 17, 1992. This work was supported in part by SIPAR - Sistemi Intelligenti SpA, Trento (Italy) and in part by CNR under contract 90.04131.CT12.

The authors are with the Dipartimento di Ingegneria dell'Informazione Università degli Studi di Parma Viale delle Scienze, 43100 Parma, Italy. IEEE Log Number 9211837.