REFERENCES

[1] A. P. Witkin, "Scale-space filtering," in *Proc. IEEE 8th Int. Joint Conf. Artificial Intelligence*, 1983, pp. 1019–1022.

[2] J. J. Koenderink, "The structure of images," *Bio. Cybern.*, vol. 50, pp. 363–370, 1984.

[3] J. Babaud and A. P. Witkin and M. Baudin and R. O. Duda, "Uniquenessof the Gaussian kernal for scale-space filtering," *IEEE Trans.. Patt. Anal. Machine Intell.*, vol. 8, pp. 26–33, 1986.

[4] A. L. Yuille and T. A. Poggio, "Scaling theorems for zero crossings," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 9, pp. 15–25, 1987.

[5] T. Lindeberg, *Scale-Space Theory in Computer Vision.* New York: Kluwer, 1994, vol. 9-7923-9418-6.

[6] D. Marr, *Vision.* San Francisco, CA: Freeman, 1982.

[7] M. H. Chen and P. F. Yan, "A multiscale approach based upon morphological filtering," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 11, pp. 694–700, 1989.

[8] P. F. M. Nacken, "Openings can introduce zero crossings in boundary curvature," *IEEE Trans. Pattern Anal. Machine Intelligence.*, vol. 16, pp. 65—658, 1994.

[9] P. T. Jackway, "Morphological scale-space," in *IEEE Proc. 11th IAPR Int. Conf. Pattern Recognition*, 1992, pp. C252–255.

[10] G. P. Dinneen, "Programming pattern recognition," in *Proc. Western Joint Computing Conf.*, 1955, pp. 94–100.

[11] J. W. Tukey, *Exploratory Data Analysis.* London, U.K.: Addison-Wesley, 1971.

[12] N. C. Gallagher and G. N. Wise, "A theoretical analysis of the properties of median filters," *IEEE Trans. Acoust. Speech, Signal Processing*, vol. 21, pp. 1136-1141, 1981.

[13] G. R. Arce and N. C. Gallagher, "State description for the root-signal set of median filters," *IEEE Trans. Acoust. Speech, Signal Processing*, vol. ASSP-30, pp. 894–902, 1981.

[14] J. Astola, P. Heinonen, and Y. Neuvo, "On root structures of median and median-type filters," *IEEE Trans. Acoust, Speech, Signal Processing*, vol. ASSP-35, pp. 1199–1201, 1987.

[15] G. R. Arce, "Multistage order statistic filters for image sequence processing," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 39, 1991.

[16] M. P. McLoughlin and G. R. Arce, "Deterministic properties of the recursive separable median filter," *IEEE Acoust., Speech, Signal Processing*, vol. ASSP-35, pp. 98–108, 1987.

[17] P. D. Wendt, "Nonrecursive and recursive stack filters and their filtering behavior," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, pp. 2099–2107, 1990.

[18] J. A. Bangham, "Data-sieving hydrophobicity plots," *Anal. Biochem*, vol. 174, pp. 694–700, 1988

[19] ——, "Properties of an improved variant on the median filter," in *Proc. Communication, Control and Signal*, 1990, pp. 1591–1597.

[20] , "Properties of a series of nested median filters, namely the datasieve," *IEEE Trans. Signal Processing*, vol. 41, pp. 31-41, 1993.

[21] J. A. Bangham and T. G. Campbell and R. V. Aldridge, "Multiscale median and morphological filters used for 2D pattern recognition," *Signal Processing*, vol. 38, pp. 387–415, 1994.

[22] J. A. Bangham, T. G. Campbell, and M. Gabbouj, "The quality of edge preservation by nonlinear filters," in *Proc. IEEE Workshop on Visual Signal Processing and Communication*, pp. 37-39, 1992.

[23] J. A. Bangham, "Using the root datasieve to recognize and locate patterns in a data sequence," in *Proc. EUSIPCO'92*, 1992, pp. 983-987.

[24] J. A. Bangham and P. D. Ling, "Analysis of recursive median sieves," UEA, pp. 1–21, UEA-SYS, Internal Publ., 1994.

[25] G. R. Arce and N. C. Gallagher, "Stochastic analysis for the recursive median filter process," *IEEE Trans. Inform. Theory*, vol. 34, 1988.

[26] J. A. Bangham, P. Chardaire, C. J. Pye, and P. D. Ling, "Multiscale nonlinear decomposition: the sieve decomposition theorem," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 18, no. 5, May 1996.

[27] J. A. Bangham, P. D. Ling, and R. W. Harvey, "Scale-space from nonlinear filters," *IEEE Trans. Patt. Anal. Machine Intell.*, vol 18, no. 4, Apr. 1996.

[28] J. A. Bangham and F. W. D. Woodhams, "Improved width measurement using line scan cameras," *SPIE*, vol. 2662, pp. 166–170, 1996.

[29] J. A. Bangham, R. Harvey, P. D. Ling, and R. V. Aldridge, "Nonlinear scale-space in many dimensions," *Computer Vision ECCV 1996*, in Lecture Notes in Computer Science, vol. 1064, pp. 189–198.

Order Statistics Learning Vector Quantizer

I. Pitas, C. Kotropoulos, N. Nikolaidis, R. Yang, and M. Gabbouj

Abstract—**In this correspondence, we propose a novel class of learning vector quantizers (LVQ's) based on multivariate data ordering principles. A special case of the novel LVQ class is the median LVQ, which uses either the marginal median or the vector median as a multivariate estimator of location. The performance of the proposed marginal median LVQ in color image quantization is demonstrated by experiments.**

I. INTRODUCTION

Research in neural networks (NN) [1], [2] is a rapidly expanding area that has attracted the attention of scientists and engineers throughout the last decade. A large variety of artificial neural networks has been developed based on a multitude of learning techniques and having different topologies [2]. One prominent example of neural networks is the learning vector quantizer (LVQ). It is an autoassociative nearest-neighbor classifier that classifies arbitrary patterns into classes using an error correction encoding procedure related to competitive learning [1]. In order to make a distinction between the (standard) LVQ algorithm and the proposed variants that are based on multivariate order statistics, the LVQ algorithm will be called *linear* LVQ algorithm hereafter.

Let us assume a sequence of vector-valued observations $\boldsymbol{x}(n) = (x_1(n), \cdots, x_p(n))^T$ where n denotes discrete-time index and p denotes the dimensionality of vector-valued observations. Let $\{\boldsymbol{w}_i(n);\ i = 1, 2, \cdots K\}$ be a set of variable $p \times 1$ reference vectors that are randomly initialized. Competitive learning tries to find the best-matching reference vector $\boldsymbol{w}_c(n)$ to $\boldsymbol{x}(n)$ (i.e., the winner) where $c = \arg\min_i \|\boldsymbol{x} - \boldsymbol{w}_i\|$ with $\|\cdot\|$ denoting the Euclidean distance between any two vectors. In the linear LVQ, the weight vectors are updated as blocks concentrated around the winner using the recursive relations [4] as follows:

$$\begin{aligned} \boldsymbol{w}_i(n+1) &= \boldsymbol{w}_i(n) + \alpha(n)[\boldsymbol{x}(n) - \boldsymbol{w}_i(n)] \quad \forall i \in \mathcal{N}_c(n) \\ \boldsymbol{w}_i(n+1) &= \boldsymbol{w}_i(n) \quad \forall i \notin \mathcal{N}_c(n) \end{aligned} \tag{1}$$

where $\alpha(n)$ is the adaptation step sequence and $\mathcal{N}_c(n)$ denotes a neighborhood set around the winner. Equation (1) implements an unsupervised *learning procedure*. In order to obtain optimal global ordering, $\mathcal{N}_c(n)$ has to be wide enough initially and to shrink monotonically with time [1], [4]. Variants of LVQ implementing supervised learning have also been proposed [4]. Several choices of the adaptation step sequence $\alpha(n)$ (called schedules) are possible [5]. The *recall procedure* of LVQ is used to determine the class $\mathcal{C}_g$ represented by $\overline{\boldsymbol{w}}_g$ with which the vector of input observations is

Manuscript received October 1, 1994; revised November 13, 1995.

I. Pitas and C. Kotropoulos are with the Department of Informatics, University of Thessaloniki, GR-540 06 Thessaloniki, Greece (e-mail: pitas@zeus.csd.auth.gr).

N. Nikolaidis is with the Department of Electrical and Computer Engineering, University of Thessaloniki, GR-540 06 Thessaloniki, Greece.

R. Yang and M. Gabbouj are with the Signal Processing Laboratory, Tampere University of Technology, SF-33101 Tampere, Finland.

Publisher Item Identifier S 1057-7149(96)04186-3.

1057–7149/96$05.00 © 1996 IEEE

most closely associated, i.e.

$$\boldsymbol{x}(n) \in \mathcal{C}_g \quad \text{if} \quad \|\boldsymbol{x} - \overline{\boldsymbol{w}}_g\| = \min_i\{\|\boldsymbol{x} - \overline{\boldsymbol{w}}_i\|\} \tag{2}$$

where $\overline{\boldsymbol{w}}_i$ denotes the weight vector of the ith neuron after the convergence of the learning procedure.

When the learning procedure reaches equilibrium, it results in a partition of the domain of input vector-valued observations called *Voronoi tessellation* [6]. This means that the input space is partitioned into regions (called Voronoi neighborhoods) bordered by hyperplanes, such that each region contains a reference vector that is the nearest neighbor to any input vector within it. Furthermore, the reference vector of a Voronoi neighborhood is the centroid, e.g., the sample arithmetic mean of all input vectors belonging to that neighborhood. From this point of view, LVQ can be classified as a method that belongs to the class of *K-means algorithms* [7].

It can easily be seen that the reference vector for each class $i = 1, \cdots, K$ at time $n+1$ is a linear combination of the input vectors $\boldsymbol{x}(j)$ $j = 0, \cdots, n$ that have been assigned to class i. Moreover, it can be shown that only in the special case of one data class, which has a multivariate Gaussian distribution and for the adaptation step sequence $\alpha(n) = 1/(n+1)$, the winner vector is the maximum-likelihood estimator of location (i.e., the arithmetic mean of the observations that have been assigned to the class). Neither in the case of multiple classes that are normally distributed nor in the case of non-Gaussian multivariate data distributions does the linear LVQ yield the optimal estimator of the cluster means. In general, linear LVQ and its variations suffer from the following drawbacks: 1) they do not use optimal estimators for obtaining the reference vectors $\boldsymbol{w}_i, i = 1, \cdots, K$ that match the maximum likelihood estimators of location for the pdf $f_i(\boldsymbol{x})$ of each class $i = 1, \cdots, K$; and 2) they do not have robustness against the outliers that may exist in the vector observations, since it is well known that linear estimators have poor robustness properties [8].

In order to overcome these problems, we propose a novel class of LVQ's that are based on order statistics (e.g., the median) which have very good robustness properties [8], [9]. In the case of LVQ's, we should rely on multivariate order statistics [10]. In this novel class, each cluster is represented by its median, which is continuously updated after each training pattern presentation.

Based on multivariate data ordering principles, we introduce the marginal median LVQ (MMLVQ), the vector median LVQ (VMLVQ) and the marginal weighted median (MWMLVQ) in Section II. The performance of the proposed MMLVQ in color image quantization is studied in Section III. Conclusions are drawn in Section IV.

II. Learning Vector Quantizers Based On Multivariate Data Ordering

The notion of data ordering cannot be extended in a straightforward way in the case of multivariate data. There is no unambiguous, universally agreeable total ordering of N p-variate samples $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N$, where $\boldsymbol{x}_i = (x_{1i}, x_{2i}, \cdots, x_{pi})^T, i = 1, \cdots, N$. The following so-called subordering principles are discussed in [9] and [10]: *marginal ordering, reduced (aggregate) ordering, partial ordering*, and *conditional (sequential) ordering*. In marginal ordering, the multivariate samples are ordered along each one of the p-dimensions as follows:

$$x_{j(1)} \leq x_{j(2)} \leq \cdots \leq x_{j(N)} \qquad j = 1, \cdots, p$$

i.e., the sorting is performed in each channel of the multichannel signal independently. The ith marginal order statistic is the vector $\boldsymbol{x}_{(i)} = (x_{1(i)}, x_{2(i)}, \cdots, x_{p(i)})^T$. Accordingly, the marginal median is the vector $\boldsymbol{x}_{\text{med}}$ defined by

$$\boldsymbol{x}_{\text{med}} = \begin{cases} (x_{1(\nu+1)}, \cdots, x_{p(\nu+1)})^T \\ \quad \text{for } N = 2\nu + 1 \\ \left(\dfrac{x_{1(\nu)} + x_{1(\nu+1)}}{2}, \cdots, \dfrac{x_{p(\nu)} + x_{p(\nu+1)}}{2}\right)^T \\ \quad \text{for } N = 2\nu. \end{cases} \tag{3}$$

It can be used in the following way in order to define the marginal median LVQ. Let us denote by $\boldsymbol{X}_i(n)$ the set of the vector-valued observations that have been assigned to class $i, i = 1, \cdots, K$ until time $n-1$. We find at time n the winner vector $\boldsymbol{w}_c(n)$ that minimizes $\|\boldsymbol{x}(n) - \boldsymbol{w}_i(n)\|, i = 1, \cdots, K$. The *marginal median LVQ* (MMLVQ) updates the winner reference vector as follows:

$$\boldsymbol{w}_c(n+1) = \text{median}\ \{\boldsymbol{x}(n) \cup \boldsymbol{X}_c(n)\}. \tag{4}$$

The median operator is given by (3). Thus, all past class assignment sets $\boldsymbol{X}_i(n), i = 1, \cdots, K$ are needed for MMLVQ. MMLVQ requires the calculation of the median of data sets of ever increasing size, as can be seen from (4). This may pose severe computational problems for relatively large n. However, for integer-valued data, a modification of the *running median algorithm* proposed by Huang *et al.* [13] can be devised to facilitate greatly median calculations by exploiting the fact that the marginal median of the already assigned samples $\boldsymbol{X}_i(n)$ is known. This algorithm leads to very large computational savings. It must be noted that, although MMLVQ employs the entire past data set for the calculation of the new weight vectors, the algorithm does not require the storage of the past data samples. Only the storage of the marginal histograms for each class is needed. The asymptotic properties of MMLVQ have been studied in [14]. It has been proven that MMLVQ outperforms the (linear) LVQ with respect to the bias in estimating the true cluster means both for a contaminated Gaussian data model as well as for a contaminated Laplacian data model. As far as the mean squared estimation error is concerned, it has been proven that MMLVQ outperforms the (linear) LVQ in the case of a contaminated Laplacian data model.

Another definition of the multichannel median is based on R-ordering principles. It is the so-called *vector median* proposed in [12]. In R-ordering, the various data $\boldsymbol{x}_i$ are ordered according to their distances from a predefined point. That is, multivariate ordering is reduced to 1-D ordering. The vector median is defined as the vector $\boldsymbol{x}_{\text{med}}$ that minimizes the L_1 error norm $d(\boldsymbol{x}_i, \boldsymbol{x}_{\text{med}}) = \Sigma_{i=1}^N\ |\boldsymbol{x}_i - \boldsymbol{x}_{\text{med}}|$ under the condition that it belongs to the set $\{\boldsymbol{x}_i, i = 1, \cdots, N\}$. In other words,

$$\sum_{i=1}^N |\boldsymbol{x}_i - \boldsymbol{x}_{\text{med}}| \leq \sum_{i=1}^N |\boldsymbol{x}_i - \boldsymbol{x}_j| \qquad j = 1, \cdots, N. \tag{5}$$

The *vector median* LVQ (VMLVQ) uses the following formula to update the winner vector $\boldsymbol{w}_c(n)$ at step n:

$$\boldsymbol{w}_c(n+1) = \text{vector median}\ \{\boldsymbol{x}(n) \cup \boldsymbol{X}_c(n)\}. \tag{6}$$

The vector median operator in the previous expression is the one defined in (5). Other distance measures (e.g., Mahalanobis distance) can be used in (5) instead of the L_1 norm as well.

Another possible extension results by employing the weighted median (WM) filter [11]. The *marginal weighted median LVQ* (MWMLVQ) can be defined as follows. Let us denote by $\boldsymbol{w}_i(n) =$

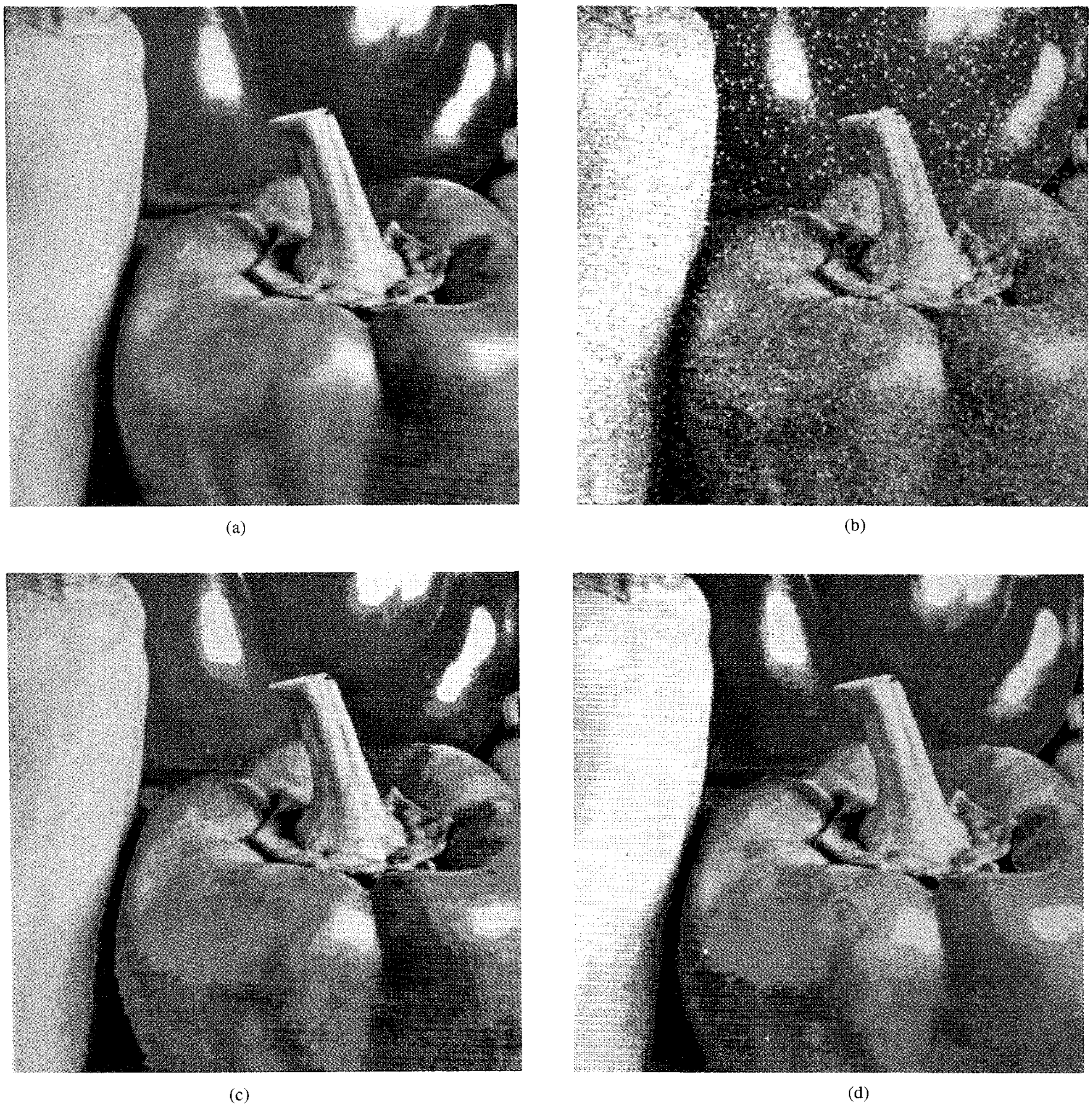

Fig. 1. Application of linear LVQ and marginal median LVQ in color image quantization in the presence of mixed additive Gaussian and impulsive noise (codebook size = 32). (a) Original image. (b) Noisy image used in the learning phase. (c) Quantized image produced by the recall phase of the linear LVQ on the original image. (d) Quantized image produced by the recall phase of the MMLVQ on the original image.

$(w_{i1}(n), w_{i2}(n), \cdots, w_{ip}(n))^T$ the winner vector, i.e., $c = i$. In MWMLVQ, the elements of the winner vector are updated as follows:

$$w_{ij}(n+1) = \text{median}\{C_{i0} \diamond x_j(n), \cdots, C_{in} \diamond x_j(0)\} \quad (7)$$

where $(C_{i0}, C_{i1}, \cdots, C_{in})^T$ is the vector of the duplication coefficients for the ith class. The duplication coefficients can be chosen in such a way so that they weigh heavily the desired section of the observation data (i.e., the new observations or the old ones). If a weight C_{il} is zero, this means that the corresponding sample $\boldsymbol{x}(n-l)$ has not been assigned to the ith class.

III. Simulations

In practice, 8 bits are used for representing each of the R, G, B components in color images. That is, 24 bits are needed for each pixel in total. Color image quantization aims at reducing the number of RGB triplets (which are 2^{24} at most) to a predefined number (e.g., 16 up to 256) of codevectors so that the image can be displayed in limited palette displays. Several algorithms have been proposed recently [15]–[18]. In the following, we shall focus on the performance of the marginal median LVQ. A set of experiments have been conducted in order to assess its performance in color image quantization and to compare it to one of the well-known vector quantization (VQ) methods such as the Linde–Buzo–Gray (LBG) algorithm [19], [20] and the linear LVQ. Moreover, we aim at studying the robustness of the codebooks (i.e., color palettes) determined by the above-mentioned VQ techniques. To this end, we have also included noisy color images as inputs to the learning phase.

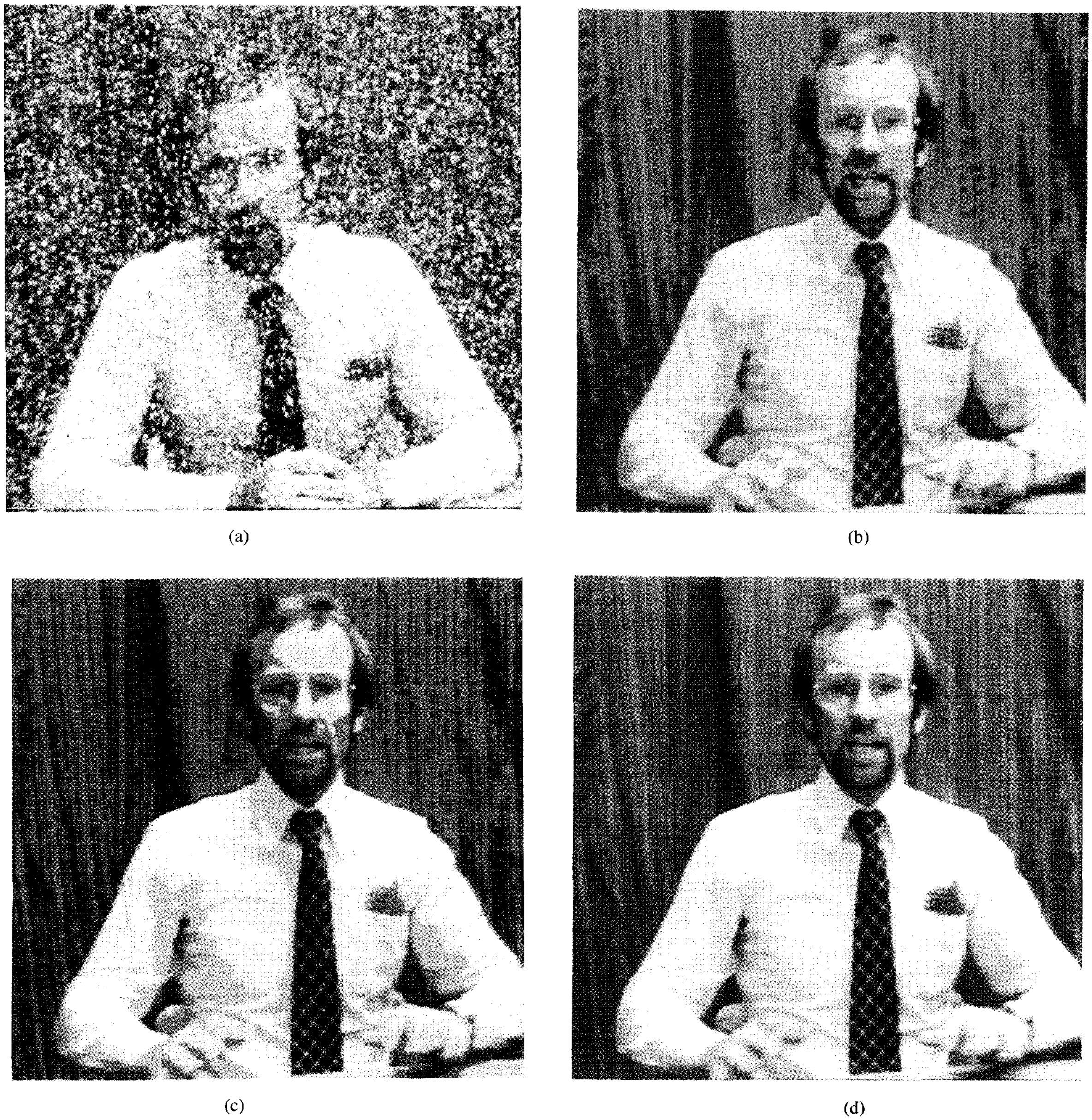

Fig. 2. Application of linear LVQ and marginal median LVQ in quantizing the 50th frame of color image sequence Trevor White in the presence of mixed additive Gaussian and impulsive noise (codebook size = 32). (a) Noisy first frame of the image sequence used in the learning phase. (b) Quantized image produced by the recall phase of the linear LVQ on the noise-free 50th frame of the image sequence. (c) Quantized image produced by the recall phase of the MMLVQ on the the noise-free 50th frame of the image sequence. (d) Original (noise-free) frame 50 of the image sequence.

Let us first define when we declare that the learning procedure in the VQ techniques included in our study has converged. During the learning phase, each VQ algorithm is applied to the training set several times. Each presentation of the training set is called training session hereafter. At the end of each training session k, the mean squared error (MSE) between the quantized and the original training patterns (i.e., RGB triplets) is evaluated as follows:

$$\mathcal{D}(k) = \frac{1}{\mathrm{card}(\mathcal{S})} \sum_{\boldsymbol{x}(i,j) \in \mathcal{S}} \|\boldsymbol{x}(i,j) - \tilde{\boldsymbol{x}}^{(k)}(i,j)\|^2 \qquad (8)$$

where $\mathcal{S}$ denotes the training set, card$(\mathcal{S})$ stands for the cardinality of the training set, $\boldsymbol{x}(i,j) = (x_R(i,j), x_G(i,j), x_B(i,j))^T$ represents the original training pattern, and $\tilde{\boldsymbol{x}}(i,j)$ is the quantized pattern. The training patterns can be obtained from the input color image, e.g., by subsampling. We decide that the learning procedure has converged if

$$\left| \frac{(\mathcal{D}(k-1) - \mathcal{D}(k))}{\mathcal{D}(k)} \right| \leq \rho \qquad (9)$$

where ρ is a small number, e.g., $\rho = 10^{-4}$. In LBG or in linear LVQ using the adaptation step sequence

$$\alpha(n) = \frac{1}{n+1} \qquad (10)$$

$\mathcal{D}(k)$ is a monotonically decreasing function, so termination rule (9) is well suited. This is not always the case with the MMLVQ or the linear LVQ using the linear schedule

$$\alpha(n) = \lambda\Big(1 - \frac{n}{M}\Big), \qquad 0 < \lambda \leq 1 \qquad (11)$$

TABLE I
PERFORMANCE OF LBG, LINEAR LVQ AND MARGINAL MEDIAN LVQ IN COLOR IMAGE QUANTIZATION, FOR SEVERAL CODEBOOK SIZES

(a) Learning phase

Codebook Size	LBG		linear LVQ adapt. step (10)		linear LVQ adapt. step (11)		MMLVQ	
	Iter.	MSE	Iter.	MSE	Iter.	MSE	Iter.	MSE
16	13	297.005	16	296.116	20	298.55	10/11	297.52
32	18	159.78	24	163.03	54	161.84	8/9	167.73
64	29	96.147	21	94.55	52	96.53	8/13	104.29
128	18	59.138	37	58.52	36	59.06	8/14	63.28
256	17	35.40	19	34.98	48	34.69	11/14	39.79

(b) Recall phase

Codebook Size	MSE			
	LBG	linear LVQ adapt. step (10)	linear LVQ adapt. step (11)	MMLVQ
16	297.114	296.116	298.55	300.95
32	168.88	172.80	171.109	175.27
64	105.844	105.106	105.74	111.16
128	69.137	68.84	69.07	71.22
256	46.11	46.46	45.95	48.30

where M is an arbitrary large number. Consequently, in the latter algorithms, the termination rule (9) should take the form $|\mathcal{D}(k-1)-\mathcal{D}(k)/\mathcal{D}(k)| \leq \rho$. In addition, the reference vectors determined by these algorithms are the ones for which $\mathcal{D}(k)$ attains its minimum value. In our comparative study, we have used as figures of merit (i) the MSE between the quantized patterns and the original ones at the end of the learning phase; (ii) the MSE at the end of the recall phase that is our ultimate goal; and (iii) the number of training sessions needed for convergence. The MSE at the end of the recall phase is defined similarly to (8) i.e.,

$$\text{MSE} = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\|\boldsymbol{x}(i,j)-\tilde{\boldsymbol{x}}(i,j)\|^2 \qquad (12)$$

where N is the number of image rows/columns.

Let us describe the selection of parameters in the different VQ techniques. All learning procedures in our study are initialized by the splitting technique used for initializing the LBG algorithm [19]. The choice of the adaptation step sequence in the linear LVQ is the most crucial for its performance. If LVQ is properly initialized and a right choice for the adaptation step sequence is made, then the experimental results indicate that LVQ will attain a performance similar to the one of LBG. Both the adaptation step sequences (10) and (11) have been considered. The adaptation step sequence (10) does not pose any further problem. For the linear schedule (11) for small codebook sizes, e.g., 16, 32, we set $\lambda = 0.003$ and $M = 10^5$. For codebook sizes 64 up to 256 we have chosen $0.03 \leq \lambda \leq 0.08$ and $M = 10^4$. In the case of MMLVQ, we have to choose an adequate odd number of samples to be assigned to each class in order to initialize the median calculation. This number was usually given a small value e.g., 5, 9, etc. Its selection does not influence the performance of the algorithm, which always converges in a similar way. That is, the MSE achieved at the end of the learning phase or the number of the training sessions is not significantly affected. Furthermore, in cases where the running algorithm proposed by Huang *et al.* [13] is applicable (e.g., in image processing), each training session of the MMLVQ requires less computation time than the one of LBG/LVQ because the learning procedure of MMLVQ does not involve any floating-point arithmetic. This is not the case with LBG and linear LVQ that require floating point operations. Having defined the parameters involved in the VQ techniques under study, we proceed to the description of the experiments.

Fig. 1(a) shows the original Pepper image of dimensions 256 × 256 with 24 b/pixel. A training set of 4096 RGB triplets has been created by subsampling the original image by a factor of four both horizontally and vertically. The performance of the LBG, the linear LVQ with adaptation step sequences (10) and (11), and the marginal median LVQ in the learning phase is listed in Table I(a). In the same table, we have included the number of iterations required for the convergence of the learning procedure of each VQ technique. For MMLVQ, two numbers are given for each entry because the training session for which the minimum is found does not coincide with the last one. The first number denotes the training session where the minimum is found and the second one the training session for which the modified termination rule employing absolute values is satisfied. The MSE at the end of the recall phase is tabulated in Table I(b). From Table I(a) and (b) it is seen that all VQ techniques attain an approximately identical performance with respect to the MSE at the end of the learning and recall phase. However, it is worth noting that MMLVQ yields close to optimal reference vectors in fewer iterations ($\simeq 10$) compared to other VQ techniques. For example, when the learning phases of LVQ and MMLVQ algorithms with 256 output neurons (last row in Table I(a)) were running on a Silicon Graphics Indy Workstation, the execution times were 37.95 s (LVQ with adaptation step (10)), 91.22 s (LVQ using adaptation step (11)) and 28.47 s (MMLVQ), respectively.

Next, we have considered the case of a noisy training set. Fig. 1(b) shows the original color image corrupted by adding mixed white zero-mean Gaussian noise having standard deviation $\sigma = 20$ and impulsive noise with probability of impulses (both positive and negative ones) $p = 5\%$ independently to each R,G,B component. A training set of 4096 noisy training patterns has been created by subsampling the noisy image of Fig. 1(b) by a factor of four both in the horizontal and vertical direction. The reference vectors determined by the learning procedure of the LBG, the linear LVQ and the MMLVQ on the noisy training set have been applied subsequently to the original image of Fig. 1(a) in order to reduce the number of RGB triplets to a

TABLE II
PERFORMANCE OF LBG, LINEAR LVQ AND MARGINAL MEDIAN LVQ IN QUANTIZING A COLOR IMAGE WHEN THE LEARNING PROCEDURE HAS BEEN DONE ON A NOISY TRAINING SET, FOR SEVERAL CODEBOOK SIZES

Codebook Size	MSE		
	LBG	linear LVQ adapt. step (11)	MMLVQ
16	639.63	658.55	584.57
32	495.58	484.35	411.20
64	329.58	386.14	330.50
128	237.09	244.175	224.66
256	149.51	153.51	143.02

TABLE III
PERFORMANCE OF THE LBG, THE LINEAR LVQ AND MARGINAL MEDIAN LVQ IN QUANTIZING SEVERAL FRAMES OF COLOR IMAGE SEQUENCE "TREVOR WHITE" WHEN THE LEARNING PROCEDURE HAS BEEN APPLIED ON THE FIRST FRAME IN THE PRESENCE OF MIXED ADDITIVE GAUSSIAN AND IMPULSIVE NOISE (CODEBOOK SIZE = 32)

Frame no.	Recall MSE		
	LBG	Linear LVQ adapt. step (11)	MMLVQ
10	289.65	313.46	240.32
50	283.27	298.84	222.38
100	295.43	312.01	242.52
120	285.27	308.36	226.46
129	282.22	307.14	224.05

predefined number. The MSE achieved at the end of the recall phase for several codebook sizes is listed in Table II. It is seen that MMLVQ outperforms the LBG and the linear LVQ for a codebook size up to 128. The quantized images produced by linear LVQ and MMLVQ for a codebook size of 32 are shown in Fig. 1(c) and (d), respectively. By comparing the latter images, the superiority of the quantized output of MMLVQ is easily deduced.

Moreover, we have considered the case of a codebook of size 32 RGB triplets that is learned from a training set of 4096 patterns extracted from a noisy frame of the color image sequence Trevor White and is applied to quantize several original frames of the same color image sequence. Fig. 2(a) shows the first frame of Trevor White corrupted by adding mixed white zero-mean Gaussian noise having standard deviation $\sigma = 20$ and impulsive noise with probability of impulses $p = 7\%$ independently to each R,G,B component. The reference vectors determined at the end of the learning procedure on the training set have been applied to quantize the tenth, 50th, 100th, 120th, and 129th frame. The MSE achieved at the end of the recall phase of both LVQ and MMLVQ is shown in Table III. It is seen that the color palette determined by MMLVQ yields the smallest MSE in all cases. The quantized images produced by the linear LVQ and the MMLVQ when frame 50 is considered are shown in Fig. 2(b) and (c). For comparison purposes, the original frame 50 is shown in Fig. 2(d). Once more it is verified that the visual quality of the quantized output of MMLVQ is higher than that of the linear LVQ.

IV. CONCLUSIONS

A novel class of learning vector quantizers has been introduced in this correspondence. These vector quantizers make use of multivariate data ordering so as to obtain multivariate estimators of location that exhibit robustness against outlying observations and erroneous classification decisions. One example of this class of LVQ's is the marginal median LVQ. Various experiments were conducted in order to test the effectiveness of the marginal median learning vector quantizer in color image quantization. The results demonstrate the superior performance of MMLVQ in comparison with the performance of classical LVQ and LBG, especially in the case of noisy data. This fact, along with its smaller execution time, makes MMLVQ a good competitor to the classical LVQ algorithm.

REFERENCES

[1] T. K. Kohonen, *Self-Organization and Associative Memory*, 3rd ed. Berlin, Germany: Springer-Verlag, 1989.

[2] P. K. Simpson, *Artificial Neural Systems*. Elmsford, NY: Pergamon, 1990.

[3] Special Issues on Neural Networks: Part I—Theory and modeling; Part II—Analysis, techniques, and applications, *Proc. IEEE*, vol. 78, pp. 1409–1680, Sept.–Oct. 1990.

[4] T. K. Kohonen, "The self-organizing map," *Proc. IEEE*, vol. 78, pp. 1464–1480, Sept. 1990.

[5] E. Yair, K. Zeger, and A. Gersho, "Competitive learning and soft competition for vector quantizer design," *IEEE Trans. Signal Processing*, vol. 40, no. 2, pp. 294–309, Feb. 1992.

[6] G. Voronoi, "Recherches sur les paralleloedres primitives," *J. Reine Angew. Math.*, vol. 134, pp. 198–287, 1908.

[7] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Stat. Prob.*, 1967, pp. 281–297.

[8] F. Hampel, E. Ronchetti, P. Rousseeuw, and W. Stahel, *Robust Statistics*. New York: Wiley, 1986.

[9] I. Pitas, A. N. Venetsanopoulos, *Nonlinear Digital Filters: Principles and Applications*. Boston, MA: Kluwer, 1990.

[10] V. Barnett, "The ordering of multivariate data," *J: Royal Stat. Soc. A*, vol. 139, part 3, pp. 318–354, 1976.

[11] D. R. K. Brownrigg, "The weighted median filter," *Comm. ACM*, vol. 27, no. 8, pp. 807–818, Aug. 1984.

[12] J. Astola, P. Haavisto, and Y. Neuvo, "Vector median filters," *Proc. IEEE*, vol. 78, no. 4, pp. 678–689, Apr. 1990.

[13] T. S. Huang, G. J. Yang, and G.Y. Tang, "A fast two-dimensional median filtering algorithm," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-27, no. 1, pp. 13–18, 1979.

[14] C. Kotropoulos, I. Pitas, and M. Gabbouj, "Marginal median learning vector quantizer," in *Signal Processing VII: Theories and Applications*, M. J. J. Holt, C. F. N. Cowan, P. M. Grant, and W. A. Sandham, Eds. New York: Elsevier, 1994, pp. 1496–1499.

[15] P. Heckbert, "Color image quantization for frame buffer display," *Comput. Graph.*, vol. 16, no. 3, pp. 297–307, July 1982.

[16] M. Orchard, and C. Bouman, "Color quantization of images," *IEEE Trans. Signal Processing*, vol. 39, no. 12, pp. 2677–2690, Dec. 1990.

[17] N. Chaddha, W.-C.Tan, and T. Meng, "Color quantization of images based on human vision perception," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Adelaide, Australia, 1994, vol. V, pp. 89–92.

[18] T.-S. Lin, and L.-W.Chang, "Greedy tree growing for color image quantization," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Adelaide, Australia, 1994, vol. V, pp. 97–100.

[19] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. 28, no. 1, pp. 84–95, Jan. 1980.

[20] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Boston, MA: Kluwer, 1992.