Vector Quantization for License-Plate Location and Image Coding

Rodolfo Zunino, *Member, IEEE*, and Stefano Rovetta, *Member, IEEE*

***Abstract*—License-plate location in sensor images plays an important role in vehicle identification for automated transport systems (ATS's). This paper presents a novel method based on vector quantization (VQ) to process vehicle images. The proposed method makes it possible to perform superior picture compression for archival purposes and to support effective location at the same time. As compared with classical approaches, VQ encoding can give some hints about the contents of image regions; such additional information can be exploited to boost location performance. The VQ system can be trained by way of examples; this gives the advantages of adaptiveness and on-field tuning. The approach has been tested in a real industrial application and included satisfactorily in a complete ATS for vehicle identification.**

***Index Terms*—Image analysis, image coding, road vehicle identification, vector quantization.**

I. Introduction

AUTOMATED vehicle identification is crucial to commercial applications such as surveillance, monitoring, billing and ticketing. In the area of automated transport systems (ATS's), license-plate recognition by visual image processing is currently the most promising approach [1]–[9], as passive sensing prevents vehicles from carrying on-board transponders. On the other hand, the high-dimensional signals involved in such a method may impose a notable computational load on an automated system.

A visual vehicle-identification system is mainly made up of a few distinct modules with specific signal-processing functions (Fig. 1). A low-level imaging module restores signal quality by noise filtering, contrast enhancement, etc.; the algorithms are usually selected according to the kind of sensor used as well as environmental conditions. As a second step, the location of interesting scene regions is attained by a segmentation process. This crucial phase may involve classical image-processing algorithms [1]–[3], genetic-algorithm techniques [4], Markov random fields [5], fuzzy sets [6], [7], or neural-network models [8], [9]. Location results feed the actual vehicle-identification module, which is generally supported by optical character recognition (OCR) methods. In some applications requiring archival facility for time-logging purposes, image-compression algorithms can also be applied to reduce storage space.

Manuscript received September 16, 1998; revised May 20, 1999. Abstract published on the Internet November 11, 1999. This work was supported in part by the Italian Ministry for University and Scientific and Technological Research (MURST).

The authors are with the Department of Biophysical and Electronic Engineering, University of Genoa, 16145 Genova, Italy.

Publisher Item Identifier S 0278-0046(00)01335-6.

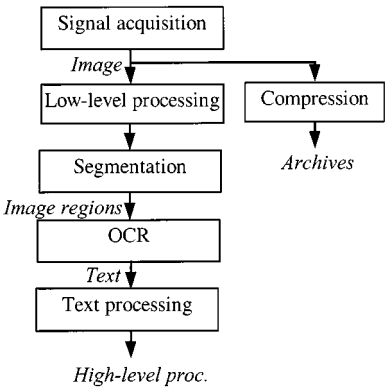


Fig. 1. Global schema of an automated license-plate recognition system.

This paper tackles the problem of license-plate location in visual signals, and presents a novel methodology that exploits vector quantization (VQ) [10] as the basic image-processing paradigm. As compared with the above-mentioned approaches, using VQ enables an ATS to support both plate location and image coding simultaneously and efficiently. The baseline of the present approach is the fact that, at high compression ratios with real-time constraints, VQ-based image coding outperforms standard methods in visual quality. The core of the proposed research lies in showing that specific and proper use of VQ methods can make location straightforward as well. In particular, a theoretical framework demonstrates that VQ image representation matches the operational requirements for the location process; then a practical algorithm for setting up VQ-based location systems is described.

As far as testing is concerned, the VQ-based location/archival method was inserted in a complete, commercial automated system for car-id applications (access control to parking areas and billing). The approach proved very satisfactory in terms of both location accuracy and coding effectiveness. Section II first briefly describes the core of VQ-based image representation and coding; then VQ-based location is theoretically justified. Section III outlines practical algorithms to set up a location system, and Section IV reports on experimental results obtained. Concluding remarks are made in Section V.

II. General Framework for VQ-Based Image Coding and Plate Location

A. VQ-Based Image Processing

Compression is the reference application area of VQ [10] in two–dimensional (2-D) signal processing [11]. The various

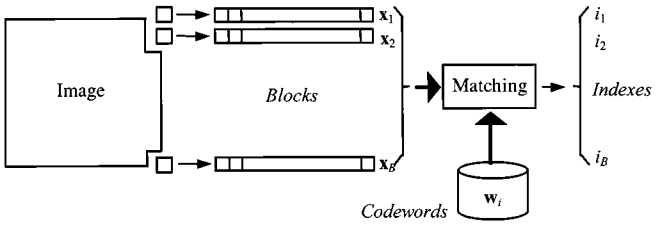


Fig. 2. Basic VQ-based schema for image coding.

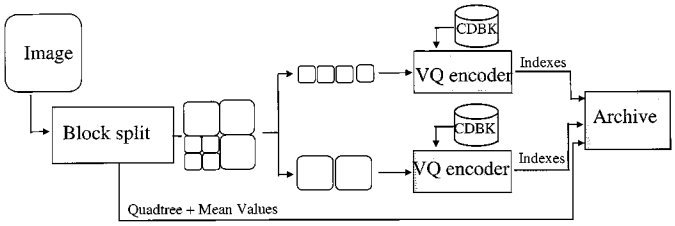


Fig. 3. Size-adaptive schema of VQ-based image compression.

approaches described in the literature can be grouped into two categories: most methods quantize signals in the frequency domain after applying transforms (e.g., discrete cosine transform) [12], [13]; other approaches apply VQ to sensor data in the pixel domain [10]. In the present context, specific features of the location process require VQ to operate according to the latter methods.

Fig. 2 shows a schematic representation of VQ-based image coding. The processed image is split into a set, X, of elementary blocks $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_B\}$, which define vectors in a data space where the quantization process exploits a predetermined, fixed "codebook" of reference vectors ("codewords") $W = \{\mathbf{w}_1, \ldots, \mathbf{w}_N\}$. The coding process associates each block $\mathbf{x}_i$ with the codeword $\mathbf{w}^*(\mathbf{x}_i)$ that optimizes a similarity criterion defined by a Minkowski metric, d_λ; the block is encoded by the codeword's index. The codeword selection process can be formalized as $\mathbf{w}^*(\mathbf{x}_i) = \arg\min_{w_n}\{d_\lambda(\mathbf{x}_i, \mathbf{w}_n)\}$. Compression results from using a codebook that is "small" as compared with the number of possible blocks. Therefore, VQ image coding is a *lossy* process, as reconstructed images differ from original ones due to quantization noise. In practice, measuring quantization noise punctually is impossible, hence, the average distortion cost, E, is evaluated as the mean-square error on the encoding process

$$E = \frac{1}{B}\sum_{i=1}^{B} d(\mathbf{x}_i, \mathbf{w}^*(\mathbf{x}_i)) = \frac{1}{B}\sum_{i=1}^{B} \|\mathbf{x}_i - \mathbf{w}^*(\mathbf{x}_i)\|^2. \quad (1)$$

The algorithm placing codewords, W, in the data space ultimately determines the effectiveness (compression ratio) and the quality (overall distortion) of a coding system. Ideally, the optimal solution to the quantization problem implies to find the codebook that minimizes the average distortion over the entire volume of the data space. This proves unfeasible in practice because one cannot know in advance the actual data distribution. Therefore, all VQ training methods use an example-driven strategy: the algorithm is given a (training) set of image blocks and yields the codeword set that minimizes distortion over the input samples; thus the empirical cost (1) is used as a Monte Carlo estimate for the actual cost. The main reason for using Euclidean metrics $(\lambda = 2)$ in measuring distortion is that, in order to minimize the cost (1), one can set up an iterative, stochastic gradient-descent algorithm that adjusts the codewords' positions whenever a sample $\mathbf{x}$ is processed

$$\Delta\mathbf{w}^*(\mathbf{x}) \propto -\eta\frac{\partial E}{\partial \mathbf{w}^*}(\mathbf{x}) \Rightarrow \Delta\mathbf{w}^* \propto 2\eta(\mathbf{x} - \mathbf{w}^*(\mathbf{x})), \ \forall \mathbf{x} \in X. \quad (2)$$

The update rule (2) is the basic common expression for codeword adjustment and can be found in all VQ training algorithms. The large variety of codebook-fitting methods proposed in the literature [14]–[16] differ in the specific strategy for applying the basic rule (2).

In the plate-location approach, knowing in advance the application domain of the coding system constitutes a great advantage that enables one to use significant training examples. Statistically speaking, this means that one can somehow mimic the statistical distribution of processed samples. Recent works have shown the possibility to use VQ to represent consistently a probability distribution of samples [17]. Such a mapping, however, minimizes a different distortion cost and reconstruction quality is no longer the primary goal. Classical mean-square-error (1) still represents the most promising approach for lossy compression applications. The possibility of sample-driven training indeed represents a crucial advantage of VQ that makes a system application adaptive and greatly improves flexibility. The research presented here adopted a novel algorithm ("Plastic Neural Gas"), which minimizes codebook size without affecting reconstruction quality, and, at the same time, guarantees the optimal exploitation of the codewords used [18].

Several technical improvements can add to the validity of the basic schema depicted in Fig. 2. *Mean residual coding* (MRC) [19] subtracts a block's mean value from the block's pixels before VQ encoding. The advantage of MRC lies in making block coding brightness independent. Cconversely, the idea that underlies *variable block size* is that uniform regions may be covered by larger blocks, whereas detail-rich image portions require smaller blocks to render visual information properly. The dynamic setting of block size [20] is typically driven by a thresholding criterion on the block variance, which triggers recursive block splitting. Clearly, the coding system must represent the structural arrangement of blocks within a whole picture; if square blocks recursively split into four equal subblocks, a quadtree [21] minimizes the necessary representation. Thus a VQ-based image-coding session (Fig. 3) yields three data structures: a quadtree for block layout, a set of mean values giving the average brightness within each block, and a corresponding set of codeword indexes to render details.

B. Using VQ for Plate Location

Most location methods disregard the actual contents of image regions in the search process and rather consider aspect features of candidate portions (e.g., edges [3], contrast [5], etc.). In some situations, however, peculiarities of acquired signals

might mislead the search; for example, particular colors or contrast settings in critical image areas might result in poorly segmented regions and erroneous location. The drawback of such approaches is that the location module often ignores whether the regions considered contain plate-pertinent information; therefore, the crucial textual analysis is remitted to succeeding steps of the image-understanding process.

The quantization principle helps overcome such limitations just because the coding process involves an implicit analysis of image contents. As a codebook is defined in the same (pixel) space as the encoded blocks, associating each block with its best-matching codeword implies a classification of the block content. The classification result may give some hint about the block content itself, in particular, it may establish whether the block is likely to cover a license plate. The overall location approach can be formalized as follows: let the image-content function be defined as $t(\mathbf{p}) = 1$ if pixel $\mathbf{p}(x, y)$ belongs to a license plate, and 0 otherwise. Then, the general location problem is equivalent to locating the image region S^* such that: $S^* = \max_S \oint_S t(\mathbf{p})\, d\mathbf{p}$. A license plate is realistically framed by a rectangular box; this assumption will be used throughout the paper, and the associated rectangular regions will be denoted by the term "stripes." Thus location is equivalent to finding:

$$S^* = \max_{x_0, y_0, x_1, y_1} \int_{x_0}^{x_1} \int_{y_0}^{y_1} t(x, y)\, dx\, dy. \tag{3}$$

The block-coding method prevents the location process from resolving the pixel level, hence a stripe's boundaries lie along the grid of blocks. The image-partitioning schema and the VQ-based approach impose a reformulation of the overall location problem for two basic reasons: 1) the integral (3) becomes a sum of contributions from blocks rather than from single pixels and 2) after VQ block classification, the limited number of codewords gives rise to an ambiguity problem: the same codeword may encode both "interesting" and "insignificant" blocks. Thus, the contribution from each block must express the average probability that the block's pixels convey plate information. The VQ-based location problem (3) can be restated as

$$S^* = \max_S \sum_{\mathbf{b} \in S} p(t = 1 \mid \mathbf{b}) \tag{4}$$

where $\mathbf{b}$ indexes the codewords associated with the blocks in region S, and $p(t = 1 \mid \mathbf{b})$ denotes the average probability that a block's pixels contain plate information, given that the block is coded by the codeword $\mathbf{b}$. The problem of selecting regions S, as expressed in (4), will be addressed in Section III. The important fact related to VQ-based location is that with each codeword a "score" is associated that estimates the corresponding probability of conveying plate information. The probability of a region will result from the contributions of all the codewords involved in the coding of the region itself. Location results directly from sorting out the highest-score region in the image. The following section will face the problem of estimating the single terms in the summation (4).

C. VQ-Based Training of Codeword Scores

Conditional probabilities $p(t = 1 \mid \mathbf{b})$ cannot be estimated analytically from the training set, as this would imply knowing the actual distribution of $\mathbf{b}$, i.e., considering all occurrences of the codeword $\mathbf{b}$ in all possible images. However, one can use Bayes' theorem and rewrite the conditional probability as

$$p(t \mid \mathbf{b}) = \frac{p(\mathbf{b} \mid t) \cdot p(t)}{p(\mathbf{b})} = \sigma(\mathbf{b}) \cdot p(t);$$
$$\sigma(\mathbf{b}) \triangleq \frac{p(\mathbf{b} \mid t)}{p(\mathbf{b})} \tag{5}$$

where the quantity $\sigma(\mathbf{b})$ is the codeword-related "score". This reformulation simplifies empirical training. When computing the score (5) associated with a codeword $\mathbf{b}$, the denominator is the overall probability of $\mathbf{b}$, and can be estimated from relative frequencies by counting the occurrences of that specific codeword. The numerator is the probability of using the codeword $\mathbf{b}$ when the encoded block is known to cover a license plate. To estimate the conditional probability, one can reconsider the images in the training set, crop the image portions holding the license plates, and keep track of the codewords used in those portions accordingly. Again, relative frequencies give the required estimates. Substituting (5) into (4) and disregarding constant terms yields an equivalent problem formulation

$$S^* = \max_S \sum_{\mathbf{b} \in S} p(t = 1 \mid \mathbf{b}) \Leftrightarrow \max_S \sum_{\mathbf{b} \in S} \sigma(\mathbf{b}). \tag{6}$$

In practice, in the final implementation, the codewords coding either license-plate blocks or insignificant image portions may get different rewards or penalties, respectively; for example, negative score can be given to those codewords that never contribute to encoding a license plate. The scoring process is computed off line on the image set *after* training the codebook for VQ compression. Each codeword is augmented by a content-dependent parameter that reflects the likelihood that the codeword may contribute to coding a license plate. The following section will integrate the VQ-based scoring mechanism with the stripe-extraction process. The training process is outlined in Fig. 4.

III. VQ-Based Location System

A. Problem Setting

When considering the method's run-time operation in practice, the system's specifications and the problem statement must be detailed, including the expected performance of the system itself and the basic assumptions underlying the overall approach. The image-processing subsystem treats 2-D signals coming from a single camera. An external vehicle-detection mechanism independently triggers a camera snapshot that yields an input picture, assuming that the position and angle of the camera are known and fixed. Input images are not requested to contain a license plate necessarily (i.e., false alarms in car detection must be rejected). License plates must respect some standard; for example, European license plates are of a constrained size and use a single font for lettering. However,

The codebook-construction algorithm

Input: training set, T, of reference images,
threshold settings $\{\tau_{32}, \tau_{16}, \tau_8\}$ and scoring constant, Π_1;
0. Set $V_{32} = V_{16} = V_8 = V_4 = W' = \varnothing$
1. (Split images $\in T$ into adaptive-size blocks $\rightarrow$ vector training sets, V_s)
For each image $I \in T$
Split I into largest-size blocks ($s = 32$) and add the blocks to $\mathbf{V}_{32}$
Repeat
For each block in $\mathbf{V}_s$, $s \in \{32, 16, 8\}$
If the block's variance $\xi^2 = \langle x^2 \rangle - \langle x \rangle^2 \geq \tau_s$ (x = block pixels)
Split the block into four square subblocks and add them to $V_{s/2}$
Remove the block from V_s
Until no blocks are split
2. (Train codewords by VQ algorithm $\rightarrow$ basic codebooks, W_s)
For each vector set, V_s, $s \in \{32, 16, 8, 4\}$
Apply the "Plastic Neural Gas" training algorithm [18] to place codebooks W_s
3. (Compute codeword frequencies)
For each block size, $s \in \{32, 16, 8, 4\}$
For each codeword $\mathbf{w}_n \in W_s$
Count q_n = the total number of blocks in V_s encoded by $\mathbf{w}_n$ ($q_n > 0$)
Normalize q_n to relative frequencies, p_n
4. (Crop from T the stripes holding plates $\rightarrow$ stripe-encoding codeword set, W')
For each image $I \in T$
Hand crop the image portion I' ("stripe") enclosing the license plate
Pick the subset of codewords $B' \subseteq \bigcup_{s=4,8,16,32} W_s$ that encode I'
Set $W' = W' \cup B'$ uniquely (i.e., a codeword appears only once)
5. (Compute scores)
For each codeword $\mathbf{w}'_k \in W'$
Count q'_k = the total number of blocks in I' that are encoded by $\mathbf{w}'_k$
Normalize q'_k to relative frequencies, p'_k
For each codeword $\mathbf{w}'_k \in W'$
Compute the score associated with codeword: $\sigma(\mathbf{w}'_k) = q'_k / p_k$
For each codeword $\mathbf{w}''_h \in \bigcup_{s=4,8,16,32} W_s - W'$
Penalize the codewords that do not encode plates: $\sigma(\mathbf{w}''_k) = -\Pi_1$
Output:
the trained and scored codebook: $W = \bigcup_{s=4,8,16,32} W_s$

Fig. 4. System training algorithm.

converting a system to a different plate standard just requires retraining. Lighting conditions may vary, although setups with constant brightness are obviously preferable. These assumptions reflect the conditions of an access-control monitoring system with controlled traffic.

B. Stripe Extraction

Although stripe selection does not differ from any classical location process, it is in fact much simpler. Stripe extraction exploits the same adaptive mechanism as adopted by VQ encoding: image regions with higher contrast and more details are mapped by smaller blocks. License plates belong to such a class of regions for the high contrast between text and background. In addition, average background information can be of great aid to stripe identification, as license plates have either a bright or dark background. Thus, the set of interesting image regions can be easily compiled by: 1) scanning the image structure (quadtree) for contiguous areas mapped by small-size blocks and 2) scanning the corresponding set of block mean values for high- (low-) brightness contiguous segments. The first criterion is actually more robust because the quadtree contrast-coding information is brightness-independent (block mean values are removed). By contrast, environmental conditions might affect the mean values of blocks; in addition, a brightness-dependent criterion might arbitrarily privilege one kind of license plate (e.g., white lettering on black plate rather than black on yellow, etc.).

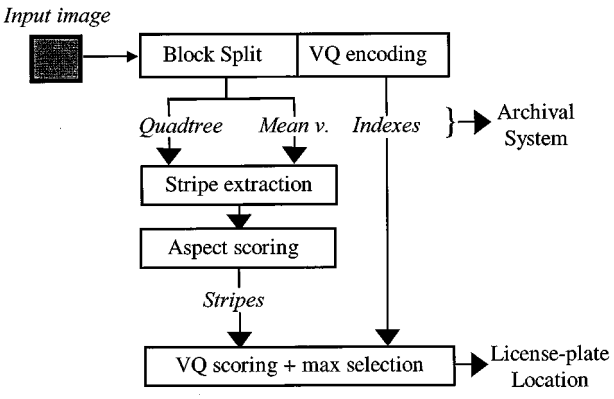


Fig. 5. Global schema of the image-compression and license-plate location systems.

The only parameter in such a process is the expected stripe size. The fact that the stripe width and height do not exhibit high variance values seems reasonable, as the eventual stripe size mostly depends on the relative sensor positioning (e.g., distance from the scene), which is assumed to be known. This implies that training a stripe extractor just consists in averaging the stripe size from examples containing license plates.

At this stage, stripe extraction can prompt an arbitrary number of selected candidates. Each stripe is scored by rewarding its consistency with the expected size; too long or too short stripes get a penalty term, Π_S, before undergoing the VQ-based location described in Section II-B.

C. Stripe Scoring and License-Plate Location

Stripe extraction simplifies the search process by working out a set of candidate regions to be considered in (6). The location module first compiles the set of codewords used to encode the blocks in the extracted stripes. This makes it possible to retrieve the related probabilities; summing up such values then characterizes each stripe accordingly. Thus the final score associated with a stripe results from two additive terms: the partial term from stripe extraction accounts for the external aspect, whereas the term summing up codeword scores takes into account the actual contents of the coded blocks. The location process eventually selects the highest-score stripe from the final sorted list. Fig. 5 presents a schematic representation of the complete system, and indirectly points out the deep-rooted link with the VQ-based compression paradigm. The run-time operation of the location process is outlined in Fig. 6.

An important property of the approach is that the process need not issue a location result for each input image. In other words, the system can also "reject" an input signal and prompt a null output, such as "no valid stripes have been located in the scene." This useful feature can be easily attained by setting a threshold for the stripe-selection mechanism at step 2): if no candidate stripe exhibits a satisfactory score, no image region meets the reliability requirement for license-plate location. It is worth stressing that this rejection ability is boosted by using VQ to inspect a stripe's content for the purpose of scoring, as a topology-oriented approach considering only aspect issues might mistake some "nice-looking" stripe.

The license-plate location algorithm

Input: trained and scored codebook, W; input image, I
threshold and constant settings $\{\tau_{32}, \tau_{16}, \tau_8, \theta_1, \theta_W, \Pi_s, e_W\}$;
0. Set $V_{32} = V_{16} = V_8 = V_4 = S = M = Z = \emptyset$
1. (VQ-compress I $\rightarrow$ quadtree Q, mean values M, codeword indexes Z)
Split I into largest-size blocks, $\mathbf{b}_n^{(32)}$ and insert the blocks in V_{32}
Repeat
For each block in V_s, $s \in \{32, 16, 8\}$
If the block's variance exceeds τ_s
Split the block, remove it from V_s,
Add generated subblocks to $V_{s/2}$
Update quadtree representation, Q
Until no blocks are split
For each block in V_s, $s \in \{32, 16, 8, 4\}$
Subtract the mean value from the block and include it in M
Find the best-matching codeword in $\mathbf{V}_s$ for the mean-residual block
Include the codeword's index in Z
2. (Extract stripes using Q and/or M $\rightarrow$ stripe set, S)
For each block, $\mathbf{b}_n^{(32)} \in$ I
Set "tentative" stripe $\mathbf{u} = \bigcup_{k \in neighb(\mathbf{b}_n)} \mathbf{b}_k^{(32)}$ ($\mathbf{u}$ encloses a square region around $\mathbf{b}_n^{(32)}$)
Count using Q the number, $u(\mathbf{u})$, of subblocks in $\mathbf{u}$
If $u(\mathbf{u}) \geq \theta$, set $S = S \cup \{\mathbf{u}\}$
3. (Compute each stripe's score $\rightarrow$ set of scored stripes, S)
For each stripe $\mathbf{u} \in S$
Retrieve using Q the set of codewords, $Z' \subseteq Z$, encoding $\mathbf{u}$
Compute the length of $\mathbf{u}$, $L(\mathbf{u})$
If $\left|\left(L(\mathbf{u}) - e_w\right)/e_w\right| > \theta_w$ Set $\sigma(\mathbf{u}) = -\Pi_s + \sum_{k \in Z'} \sigma(\mathbf{w}_k)$
else Set $\sigma(\mathbf{u}) = \sum_{k \in Z'} \sigma(\mathbf{w}_k)$
4. (Set up a list of candidates)
Sort S in descending order according to $\sigma(\mathbf{u})$
Output: stripe set, S

Fig. 6. Outline of VQ-based license-plate location.

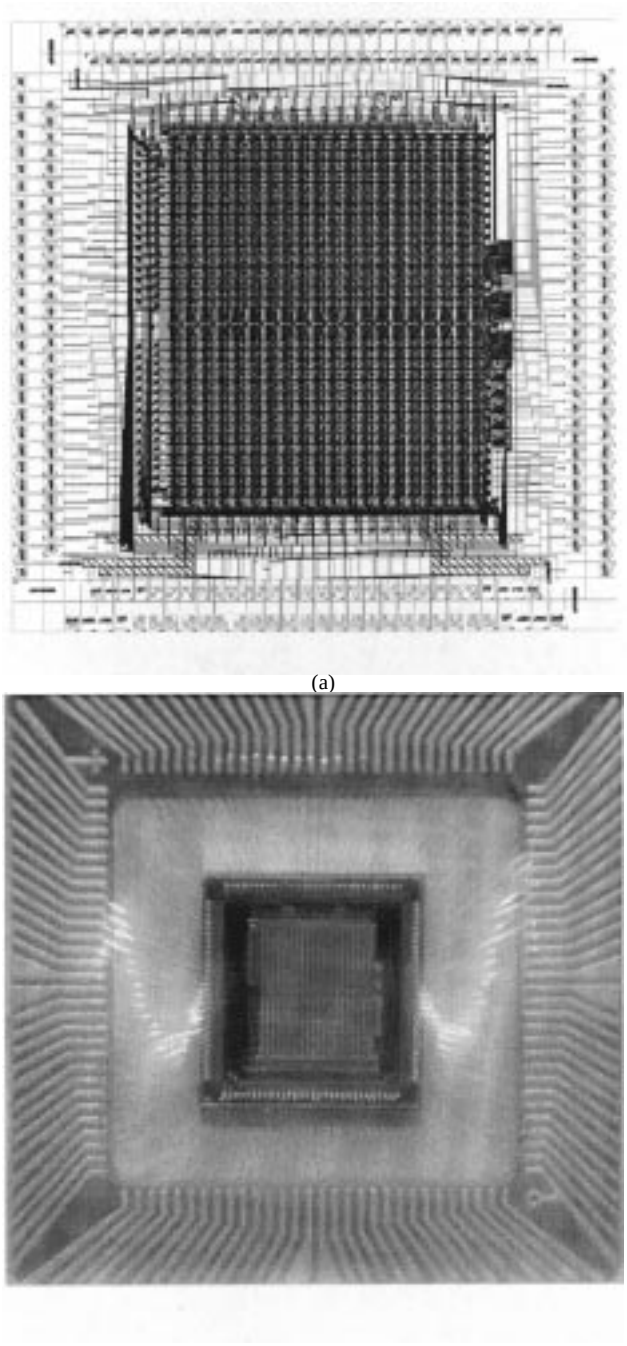
D. Performance Acceleration Issues

VQ-based methodologies typically bring about high computational costs in the search for the best-matching codeword, which can involve a large number of sample-codeword distances in both training and test. Therefore, any VQ application-oriented approach provides some means to keep execution timings under control. In the literature, the problem has been tackled from different perspectives.

Some methods decrease the computational cost of evaluating a single distance, and may set some bounds to the metrics involved [22]–[24]. Other approaches arrange codewords in a hierarchical codebook organization [25] that is faster to scan. Although, a partial codebook search should not ensure optimality of encoding performance, yet the distortion results reported in the literature show that the latter methods can profitably replace a strategy based on an exhaustive search.

From a hardware-acceleration point of view, research on efficient implementation leads to specific VQ models that are mapped directly into hardware architectures [26]. Most research, however, mainly concerns the hardware (HW) support of training algorithms [27], [28]. The few application-oriented approaches proposed are mostly used in the area of image processing [29]–[31] and provide on-chip training facility. A recent digital HW system featuring a strong practical bias for pattern recognition applications supports an exhaustive search for nearest-neighbor classification [32].

The image-processing application described in the present paper can certainly benefit from any of the above algorithmic enhancements [33]. On the other hand, hardware acceleration seems to represent the most promising approach to cutting computational costs. As input signals need not be transformed into any frequency domain, an HW-supported quantization in the pixel domain becomes cost-effective. In addition, a hardware architecture matches the arithmetical nature of problem (1) by fully exploiting the parallelism inherent in analog VLSI architectures.



(a)

(b)

Fig. 7. The VQ-encoding chip. (a) Internal layout. (b) Chip bonding.

For such reasons, a dedicated VLSI chip for VQ support of image coding has been designed and fabricated. The chip (1μ ES2 analog technology) processes 16-dimensional input samples and stores up to 64 codewords; the clock rate is 1 MHz. The detailed device specifications are given in [34]; Fig. 7 presents the schematic layout of the chip and a photograph of the device.

IV. Experimental Results

A. Application Environment

The location module was included in a complete ATS used to control vehicle access to a large area. The ATS was provided by a leading Italian firm in the areas of image understanding and optical character recognition, and supported vehicle identification, access grant, and billing in a parking area in Madrid, Spain.

The location system processed input fields (i.e., interlaced half frames) from a standard camera; the input signals were converted into digital pictures holding 768×256 pixels with 256 gray-scale levels (8 bpp). The system did not need to deinterlace a complete frame from two fields due to real-time constraints, as both the picture quality and the location performance resulting from the field signals were satisfactory. The output of the location process was fed into a suitably developed OCR module for license-plate interpretation [35], which triggered database management and further billing processing.

The current PC-based C++ implementation ran under Windows NT. An image-understanding cycle (acquisition, location, OCR) took about 200 ms on a standard Intel Pentium board running at 200 Mhz. This performance proved satisfactory and cost effective for the specific application considered. With the HW accelerator chip, the worst-case coding time for the most complex quadtree (i.e., when the size of all subblocks is 4×4) drops to 13 ms and becomes comparable with the signal-acquisition timing. The OCR timing after text location is negligible, hence HW acceleration can enhance speed performance by about one order of magnitude.

The training process of the overall system involved just ten images acquired under different environmental conditions. The final codebook included 256 codewords for each block size and was adapted using the algorithm presented in [18]. Although training proceeded off line and the related timings did not affect the system's performance, the codebook-fitting process was completed in a few minutes on the PC architecture, hence also on-line training can be envisioned if requested.

B. Image-Compression Performance

The relatively large size of the processed pictures is the basic reason for using VQ in real-time image compression. The analysis of VQ as a compression paradigm is beyond the scope of this paper, so we refer the reader to the vast specific literature on this subject [10], [11]. As far as the present discussion is concerned, the constraints imposed on the image-coding method can be summarized as follows: 1) the compression ratio must be high (i.e., $\mathrm{Cr} \geq 40$) to optimize storage in the archival system; 2) the quality of decompressed pictures must be acceptable for the application considered, that is, vehicle details must be clearly understandable and no ambiguities concerning the license-plate area must arise.

In this section, a comparison of the proposed VQ-based schema with the standard JPEG compression algorithm is made. Experimental evidence shows that the latter performs effectively at low compression ratios, whereas its performance is unacceptable at high compression ratios. Conversely, VQ suffers from blockiness effects at high bit rates, but it outperforms other related approaches at ratios higher than 30.

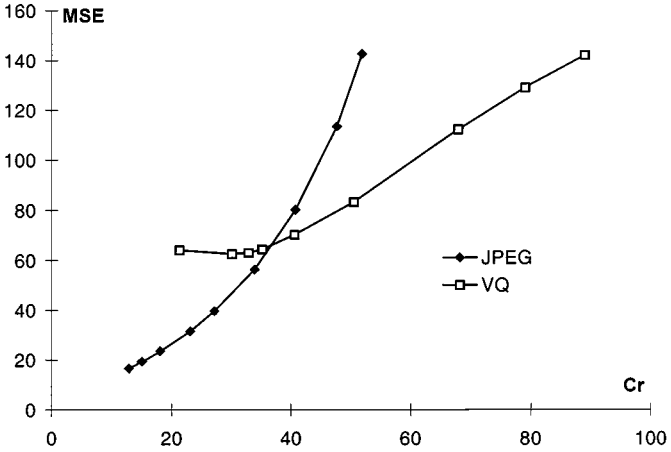


Fig. 8. Comparison of VQ and JPEG compression qualities. The VQ schema performs better at higher compression ratios (typically, $\mathrm{Cr} > 35$).

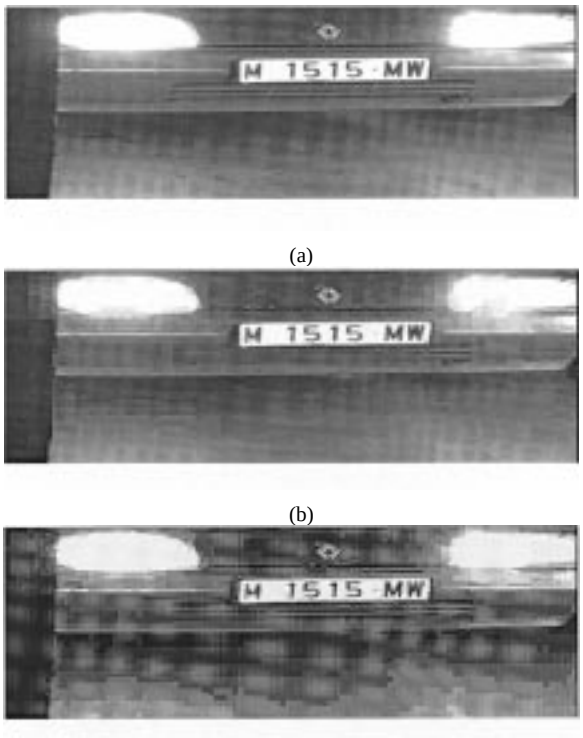


Fig. 9. Visual comparison of VQ and JPEG compression for picture archival purposes ($\mathrm{Cr} = 50$). (a) Original picture. (b) VQ-compressed picture. (c) JPEG-compressed picture.

This property is quantitatively confirmed by the graph in Fig. 8, which compares VQ with JPEG in terms of average image quality (mean-square error) versus compression ratio. However, a reliable assessment of a method's qualitative performance requires visual inspection of compressed images. To this end, Fig. 9 presents the compression performance of a sample picture at $\mathrm{Cr} = 50$, and proves the superior performance of the VQ-based method.

C. Sample of the System Operation

This section presents a complete example of the system's operation; the crucial steps of the location process are highlighted

Fig. 10. Input picture (digitized PAL field, 768×256, 8 bpp).

Fig. 11. Result of adaptive block-size decomposition.

Fig. 12. Compressed picture for archival storage.

Fig. 13. Candidate stripes extracted from Fig. 11 and their associate aspect scores.

by illustrating intermediate results. Fig. 10 shows an input image; the result of quadtree decomposition after adaptive block splitting and mean-values extraction is given in Fig. 11. VQ block coding completes image compression for picture archival storage; the image compressed at $\mathrm{Cr} = 40$ is displayed in Fig. 12.

In the actual license-plate location process, the stripe-search module first extracts candidate regions. In the present application, a detail-centered approach using quadtree information was adopted as described in Section III-B, mainly thanks to its brightness-independent performance. This criterion locates picture regions with richer detail information: they correspond to quadtree sections with many contiguous small blocks. From the image in Fig. 11, five stripes can be extracted that are presented in Fig. 13. The "spatial" score assigned to its each stripe relates to its length and rewards its consistency with the expected best-stripe length.

The succeeding content-dependent analysis retrieves, for each stripe, the codewords coding the involved blocks, and sums up the associate scores. These results add to the scores assigned by the stripe extractor, and end up with the final sorted list of candidates. Table I presents numerical results and an updated list of stripes.

TABLE I
SCORES ASSIGNED TO CANDIDATE STRIPES

	Spatial	*VQ*	*Final*
Stripe 1	-100	-1.66	-101.66
Stripe 2	0	-1.46	-1.46
Stripe 3	-100	-1.80	-101.80
Stripe 4	0	+0.99	+0.99
Stripe 5	0	-2.21	-2.21

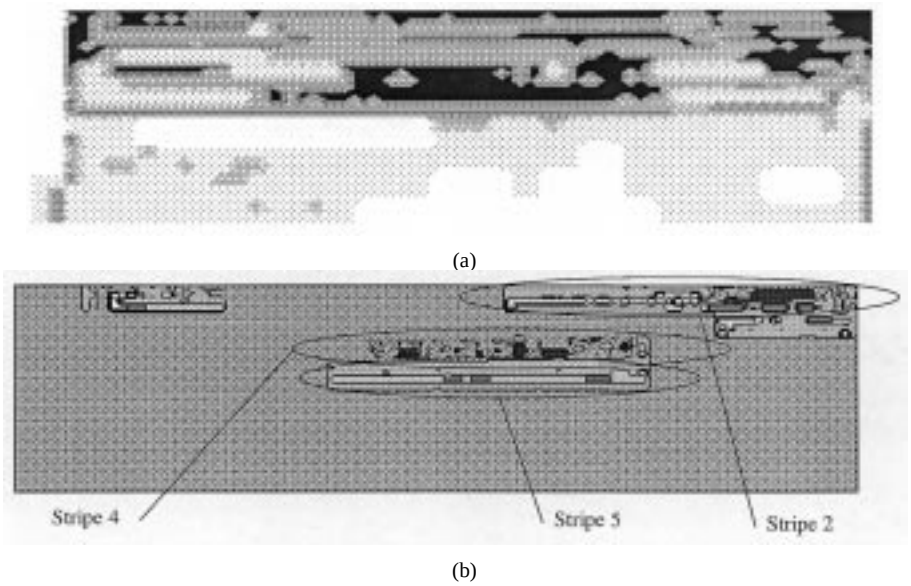
The reliability of the most promising candidate is finally evaluated: the thresholding mechanism discards the result of the entire process if no suitable stripe with a sufficiently high score has been detected. In the practical application described here, this threshold was set to zero, hence, a location process issuing only negative scores prompts a rejection output.

The final result of the overall location process consists in the top position on the sorted list, which enters the succeeding OCR module. In the example, stripe 4 is correctly prompted as the license-plate stripe. It is worth stressing, however, that placing the correct stripe as the best guess is not a strict requirement for the complete system, as OCR modules typically accept a few areas for the text-search process. In the industrial OCR used for the present application, a set of at most two stripes give a valid OCR input.

The presented example can also clarify visually the actual effect of VQ coding on the location performance. Fig. 14(a) presents visually the quadtree-driven image structure; darker areas are encoded by smaller blocks. This picture highlights the major drawback of aspect-based plate location: the license-plate area emerges but cannot be discriminated from the other stripes, which are equally likely candidates. Fig. 14(b) displays the map of blocks *after* VQ-based scoring; darker regions now indicate higher scores. This picture clearly shows how codeword labeling changes region scoring and biases decisions; for example, the different average colors associated with stripe 4 and the other discarded candidates (stripes 2 and 5) now witness correct plate location.

D. Location Performance

The location approach underwent a thorough validation process aimed at assessing its performance on a statistical basis. For the experiments, a test set was used that included more than 300 pictures, taken under quite different environmental conditions (e.g., brightness, sensor positioning, etc.). In the case of the OCR module adopted, a location process is considered successful if the license plate appears in the first two stripes on the sorted list. Therefore, the experimental verification reports on results accordingly and distinguishes the different possibilities; a location error is detected when the license plate does not show up in either of the two best-guess candidates.



(a)

(b)

Fig. 14. Visual representation of the image-region mapping. (a) Spatial-scoring map: darker areas correspond to smaller blocks. (b) VQ-scoring map: darker areas mark higher-score regions.

TABLE II
STATISTICAL SUMMARY OF LOCATION RESULTS

License plate in the best guess	87.6%
License plate in the second guess	10.4%
Location error	2%

Results are summarized in Table II; they indicate that the overall system performance is quite satisfactory, as it exhibits a 2% error rate. This performance meets industrial requirements, as the specific OCR process adopted can detect the absence of text in a stripe and then trigger an exceptional type of processing. In practice, such critical pictures are stored in a special file for delayed visual inspection.

These results compare favorably with those obtained by related approaches in the literature. For example, the overall error rate is the same as reported in [3], where a visual testbed of about the same size has been used. An advantage of the present methodology lies in its inherent robustness to variations in the acquisition process; the VQ encoding mechanism does not require specific and possibly sensitive image-processing algorithms (e.g., edge detection [3]). Conversely, the complexity of a VQ system's training process is usually lower than that required to reach convergence in genetic algorithm training [5], [4] or Markov random field tuning [5].

When reconsidering the assumptions underlying the overall methodology described in Section III-A, one might observe that the fixed and known positioning of the sensor ultimately imposes a constraint on the expected size of a plate in an image and gives rise to some issues worth discussing. First of all, the method's performance becomes dependent on the plate size. In fact, when the relative camera position with respect to traffic is considerably changed, the system needs retraining in order to adjust both the splitting parameters (for proper quadtree generation) and the expected stripe size (for aspect scoring). If codebook generation by the VQ-training algorithm eventually proves necessary, too, this can be accomplished on site at a limited cost.

Secondly, the information about the plate location might theoretically be obtained directly by searching for small-size blocks that are lying in the expected stripe size in the mapped image structure. This basic method, however, would deem equivalent all image portions fulfilling aspect-ratio conditions, whereas VQ codeword scoring makes it possible to discriminate effectively among most likely candidates.

The aforesaid feature (partially) solves another crucial issue concerning the integration of the location and OCR modules; the described procedure will also find any other feature in the image similar to a license plate. In fact, this is a problem with all image-processing methods, and solving it by remitting decisions to the next OCR step could degrade the advantage of a one-step method. The VQ-scoring mechanism helps overcome this drawback by sorting stripes according to probabilities [8]. Clearly, any text in the image with the same features as a license plate can mislead the location process, although a careful codebook training should make this event less and less likely. In the proposed system, the fact that the correct stripe is mostly one of the first two candidates confirms the method's robustness against the above-mentioned chance.

V. CONCLUSION

This paper has presented a novel approach to the license-plate location problem in an ATS. VQ-based image coding makes it possible to obtain image compression for archival purposes and, at the same time, to support the location process. The main advantage of using VQ for image representation is that a quadtree representation by the specific coding mechanism can give a system some hints about the contents of image regions. The core of the overall research, therefore, consists in fully exploiting such information to boost location performance. The described method attains this goal by training the coding + location modules accordingly; the possibility of example-driven on-field ad-

justment is a crucial advantage of the VQ-based approach, as it allows system flexibility and adaptiveness.

The reported results are promising, although a better comparison would involve a larger database. Experience indicates that thousands of images must be processed to obtain representative performance parameters, which actually requires an industrial application in the field. This step will be accomplished as soon as the HW-acceleration device is integrated in the overall system.

ACKNOWLEDGMENT

The authors wish to thank A. M. Colla, E. Ottaviani, and all the staff of the Research Department, Elsag Bailey SpA, for their valuable assistance in developing the method and for providing the application setup. They also gratefully acknowledge the cooperation of R. Carassale, who provided an effective software environment.

REFERENCES

[1] P. Comelli, P. Ferragina, M. N. Granieri, and F. Stabile, "Optical recognition of motor vehicle license plates," *IEEE Trans. Veh. Technol.*, vol. 44, pp. 790–799, Nov. 1995.

[2] H. Onoue and M. Shiono, "A license number plate extraction in a car image and recognition of all characters containing Kanji," *Trans. Inst. Electron. Inf. Commun. Eng.*, vol. J77D-II, no. 3, pp. 483–492, 1994.

[3] A. Postolache and J. Trecat, "Pyramidal approach to license plate segmentation," *J. Electron. Imag.*, vol. 5, no. 3, pp. 402–409, 1996.

[4] K. S. Kyoon, K. D. Wook, and K. H. Joon, "A recognition of vehicle license plate using a genetic algorithm based segmentation," *Proc. IEEE Int. Conf. Image Processing*, vol. 2, pp. 661–664, 1996.

[5] Y. T. Cui and Q. Huang, "Extracting characters of license plates from video sequences," *Mach. Vis. Applicat.*, vol. 10, no. 2, pp. 308–320, Mar. 1998.

[6] J. A. G. Nijhuis and M. H. T. Brugge *et al.*, "Car license plate recognition with neural networks and fuzzy logic," *Proc. IEEE Int. Conf. Neural Networks*, vol. V, pp. 2232–2236, 1995.

[7] S. Abe and R. Thawonmas, "A fuzzy classifier with ellipsoidal regions," *IEEE Trans. Fuzzy Syst.*, vol. 5, pp. 358–368, Aug. 1997.

[8] S. Draghici, "A neural network based artificial vision system for license plate recognition," *Int. J. Neural Syst.*, vol. 8, no. 1, pp. 113–126, 1997.

[9] M. Raus and L. Kreft, "Reading car license plates by the use of artificial neural networks," *Proc. 38th Midwest Symp. Circuits and Systems*, vol. 1, pp. 538–541, 1996.

[10] R. M. Gray, "Vector Quantization," *IEEE ASSP Mag.*, vol. 1, pp. 4–29, Apr. 1984.

[11] N. Nasrabadi and R. King, "Image coding using vector quantization: A review," *IEEE Trans. Commun.*, vol. 36, pp. 957–971, Aug. 1988.

[12] K. R. Rao and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, Applications*. San Diego, CA: Academic, 1990.

[13] C. Amerijck, M. Verleysen, P. Thissen, and J. D. Legat, "Image compression by self-organizing Kohonen map," *IEEE Trans. Neural Networks*, vol. 9, pp. 503–507, May 1998.

[14] Y. Linde, A. Buzo, and R. M. Gray, "An Algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COM-28, pp. 84–95, Jan. 1980.

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

[16] T. Martinetz, S. G. Berkovich, and K. Schulten, "'Neural Gas' network for vector quantization and its application to time-series prediction," *IEEE Trans. Neural Networks*, vol. 4, pp. 558–569, July 1993.

[17] M. M. Van Hulle, "Kernel-based equiprobabilistic topographic map formation," *Neural Computation*, vol. 10, pp. 1847–1871, Dec. 1998.

[18] S. Ridella, S. Rovetta, and R. Zunino, "Plastic algorithm for adaptive vector quantization," *Neural Comput. Applicat.*, vol. 7, no. 1, pp. 37–51, 1998.

[19] R. L. Baker and R. M. Gray, "Differential vector quantization of achromatic imagery," *Proc. Int. Picture Coding Symp.*, pp. 105–106, 1983.

[20] P. Strobach, "Quadtree-structured recursive plane decomposition coding of images," *IEEE Trans. Signal Processing*, vol. 39, pp. 1380–1397, June 1991.

[21] J. Vaisey and A. Gersho, "Variable block-size image coding," in *Proc. ICASSP'87*, Apr. 1987, pp. 1051–1054.

[22] J. S. Pan and K. C. Huang, "A new vector quantization image coding algorithm based on the extension of the bound for Minkowski metric," *Pattern Recognit.*, vol. 31, no. 11, pp. 1757–1760, 1998.

[23] E. Vidal, "An algorithm for finding nearest neighbors in (approximately) constant average time," *Pattern Recognit. Lett.*, vol. 4, no. 1, pp. 145–157, Jan. 1986.

[24] C. H. Lee and L. H. Chen, "Fast closest codeword search algorithm for vector quantization," *Proc. Inst. Elect. Eng.—Vision Image Signal Processing*, vol. 141, no. 3, pp. 143–148, 1994.

[25] R. Jain, A. Madisetti, and R. L. Baker, "An integrated circuit design for pruned tree-search vector quantization encoding with an off-chip controller," *IEEE Trans. Circuits. Syst. Video Technol.*, vol. 2, pp. 147–158, June 1992.

[26] K. Tsang and B. W. Y. Wei, "A VLSI architecture for a real-time code book generator and encoder of a vector quantizer," *IEEE Trans. VLSI Syst.*, vol. 2, pp. 360–364, Sept. 1994.

[27] Y. He and U. Çilingiroğlu, "A charge based on-chip adaptation Kohonen neural network," *IEEE Trans. Neural Networks*, vol. 4, pp. 462–468, May 1993.

[28] D. Macq, M. Verleysen, P. Jespers, and J. D. Legat, "Analog implementation of a Kohonen map with on-chip learning," *IEEE Trans. Neural Networks*, vol. 4, pp. 456–461, May 1993.

[29] W. C. Fang, B. J. Sheu, O. T. C. Chen, and J. Choi, "A VLSI neural processor for image data compression using self-organization networks," *IEEE Trans. Neural Networks*, vol. 3, pp. 506–518, May 1992.

[30] K. Dezhgosha, M. M. Jamali, and S. C. Kwatra, "A VLSI architecture for real-time image coding using a vector quantization based algorithm," *IEEE Trans. Signal Processing*, vol. 40, pp. 181–189, Jan. 1992.

[31] W. C. Fang, C. Y. Chang, B. J. Sheu, O. T. C. Chen, and J. C. Curlander, "VLSI systolic binary tree-searched vector quantizer for image compression," *IEEE Trans. VLSI Syst.*, vol. 2, pp. 33–44, Jan. 1994.

[32] A. Lipman and W. Yang, "VLSI hardware for example-based learning," *IEEE Trans. VLSI Syst.*, vol. 5, pp. 320–328, Sept. 1997.

[33] F. Ancona, S. Rovetta, and R. Zunino, "Efficient technique for implementing an image-compression neural algorithm on concurrent-multiprocessor architectures," *Int. J. Eng. Applicat. Artif. Intell.*, vol. 10, no. 6, pp. 573–580, 1997.

[34] S. Rovetta and R. Zunino, "Efficient training of neural gas vector quantizers with analog circuit implementation," *IEEE Trans. Circuits Syst. II*, vol. 46, pp. 688–698, June 1999.

[35] F. Ancona, A. M. Colla, S. Rovetta, and R. Zunino, "Implementing probabilistic neural networks," *Neural Comput. Applicat.*, vol. 5, no. 5, pp. 152–159, Jan. 1997.

Rodolfo Zunino (S'90–M'90) received the Laurea degree in electronic engineering from the University of Genoa, Genoa, Italy.

From 1986 to 1995, he was a Research Consultant with the Department of Biophysical and Electronic Engineering, University of Genoa, where he is currently an Assistant Professor, teaching industrial electronics. His main scientific interests include electronic systems for neural networks, distributed control, and methods for data representation and processing.

Prof. Zunino is a member of the Society of Motion Picture and Television Engineers, Inc.

Stefano Rovetta (M'98) received the Laurea degree in electronic engineering and the Ph.D. degree in models, methods, and tools for electronic and electromagnetic systems from the University of Genoa, Genoa, Italy, in 1993 and 1997, respectively.

He is currently a Postdoctoral Researcher with the Electronic Systems and Networking Group, Department of Biophysical and Electronic Engineering, University of Genoa. He is also an Invited Professor of Operating Systems, University of Siena, and teaches electronics at the University of Genoa. His research interests include electronic circuits and systems and neural network theory, implementation, and applications.