VISUAL LOCATION OF LICENSE PLATES BY VECTOR QUANTIZATION

Rodolfo Zunino and Stefano Rovetta

DIBE - Dept. of Biophysical and Electronic Engineering
University of Genoa
Via all'Opera Pia 11a - 16145 Genova - Italy
e-mail: { rovetta, zunino } @dibe.unige.it

ABSTRACT

The paper describes a methodology based on Vector Quantization (VQ) to support visual vehicle identification: license plate location is the specific task involved by VQ-based image coding. Using VQ yields superior picture compression for archival purposes and supports effective location at the same time. VQ encoding can give some hints about the contents of image regions; such information is exploited to enhance location performance. Training the VQ system by examples gives the advantage of adaptive on-field tuning. The approach has been tested in a real industrial application and included satisfactorily in an ATS for vehicle identification.

1. INTRODUCTION

A visual vehicle-identification system in mainly made up of a few distinct modules with specific signal-processing functions [1-3]: 1) a low-level imaging module restores signal quality by application-specific techniques; 2) the location of interesting scene regions is attained by a segmentation process; 3) location results feed the actual vehicle-identification module including 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.

This paper tackles the problem of license plate location in visual signals, and presents a novel methodology that exploits Vector Quantization (VQ) [4] as the basic image-processing paradigm. As compared with related approaches, using VQ enables an ATS to support both plate location and image coding simultaneously and efficiently. The baseline is that VQ-based image coding gives satisfactory visual quality at high compression ratios; the proposed research shows theoretically and practically that specific and proper use of VQ methods can make location straightforward as well.

2. USING VQ FOR PLATE LOCATION AND IMAGE CODING

2.1 Vector quantization techniques

Compression is the reference application area of Vector Quantization [4] in 2-D signal processing [5]. In the present context, VQ-based image coding operates in the pixel domain. The processed image is split into elementary blocks $\{\mathbf{x}_1,...,\mathbf{x}_B\}$, which define vectors in a data space where the quantization process exploits a predetermined, fixed "codebook" of reference vectors ("codewords") $\{\mathbf{w}_1,...,\mathbf{w}_N\}$. The coding process associates each block $\mathbf{x}_i$ with the codeword $\mathbf{w}^*(\mathbf{x}_i)$ that optimizes a similarity criterion; the block is encoded by the codeword's index. The Euclidean distance usually measures the distortion of the block-codeword matching. 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. The research presented here adopted a novel training algorithm specifically designed to place codewords optimally and minimize codebook size [6].

Mean Residual Coding (MRC) subtracts a block's mean value from the block's pixels before VQ encoding; MRC makes block coding brightness-independent. A *variable block size* enhances compression: as not all image regions convey the same amount of information, uniform regions may be covered by larger blocks, whereas detail-rich image portions require smaller blocks. The dynamic setting of block size is driven by a thresholding mechanism on the variance of pixel values within the block. If square blocks are used and each block-splitting yields four subblocks, a quadtree minimizes the structural arrangement of blocks within a whole picture. The result of VQ-based image-coding (Fig.1) includes three data structures: a quadtree ruling block layout, a set of mean values giving the average brightness within each block, and a corresponding set of codeword indexes to render details.

2.2 Using VQ for plate location

Most location methods consider aspect features of candidate portions (e.g., edges [2], contrast [1], etc.). The drawback of such approaches is that the location module often ignores

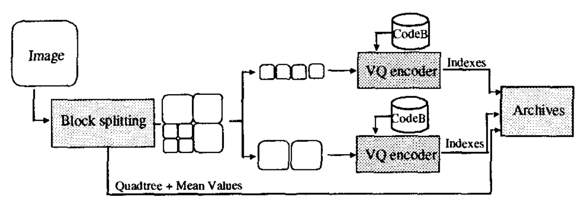


Fig. 1 - Schema of size-adaptive image compression

0-7803-5471-0/99/$10.00©1999 IEEE

whether the regions considered contain plate-pertinent information so that 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. A codebook is defined in the same (pixel) space as the encoded blocks, hence 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. A license plate can be realistically assumed to be framed by a rectangular box; in the following, the associated rectangular regions will be denoted by the term "stripes". The general location problem is equivalent to locating the image region S^* that satisfies:

$$S^* = \max_{x_0,y_0,x_1,y_1} \int_{x_0}^{x_1}\int_{y_0}^{y_1} t(x,y)\,dx\,dy \qquad (1)$$

where $t(x,y)$, is equal to 1 if location (x,y) belongs to a license plate, and is zero otherwise.

In size-adaptive image coding, a stripe's boundaries lie along the grid of blocks. Such an image-partitioning schema and the VQ-based approach impose a reformulation of the overall location problem: First, the integral (1) becomes a sum of contributions from blocks rather than from single pixels. Secondly, when considering the result of VQ block classification, the limited number of codewords gives rise to an ambiguity problem: the same codeword may encode both "interesting" and "insignificant" blocks. The contribution from each block must express the average probability that the block's pixels convey plate information. The VQ-based location problem (1) is restated as

$$S^* = \max_{S} \sum_{\mathbf{b}\in S} p(t=1|\mathbf{b}) \qquad (2)$$

where $\mathbf{b}$ indexes the codewords associated with the blocks in region S, whereas $p(t=1|\mathbf{b})$ denotes the average probability that a block's pixels contain plate information, given that the block is coded by codeword $\mathbf{b}$. In VQ-based location, 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 codewords involved in the coding of the region itself. Location results directly from sorting out the highest-score region in the image.

2.3 VQ-based training of codeword scores

Conditional probabilities in (2) cannot be estimated directly from a training set, as this would imply knowing the actual distribution of $\mathbf{b}$, i.e., considering all occurrences of codeword $\mathbf{b}$ in all possible images. However, one can use Bayes' theorem and, disregarding constant terms, derive a new problem formulation:

$$S^* = \max_{S} \sum_{\mathbf{b}\in S} p(t=1|\mathbf{b}) = \max_{S} \sum_{\mathbf{b}\in S} \frac{p(\mathbf{b}|t=1)}{p(\mathbf{b})} \qquad (3)$$

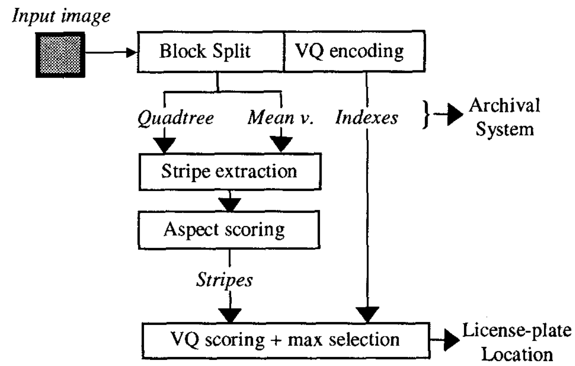


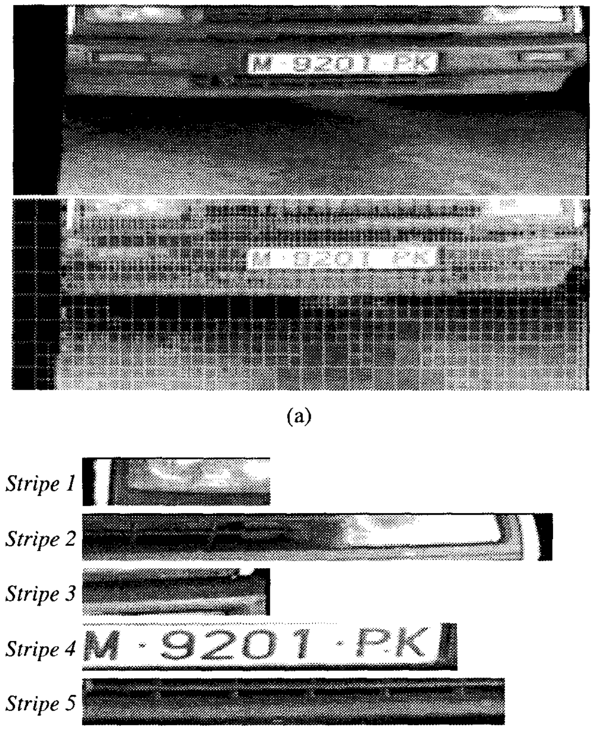
Fig.2 - Global schema of the image-compression and license-plate location systems

which makes empirical training viable. The denominator of each term in summation (3) is the overall probability of codeword $\mathbf{b}$, and can be evaluated from relative frequencies by counting the occurrences of $\mathbf{b}$. The numerator is the probability of using codeword $\mathbf{b}$ when the encoded block is known to cover a license plate. In order 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 used codewords accordingly. Again, relative frequencies will give the required estimates.

In practice, the final implementation may also differentiate between scoring terms: therefore, the codewords coding either license-plate blocks or insignificant image portions may get different rewards or penalties, respectively. 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.

2.4 Stripe extraction

Stripe extraction is the system process selecting the candidate regions to be considered in (2). It 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. Thus the set of interesting image regions can be easily compiled by searching the image structure (quadtree) for contiguous areas mapped by small-size blocks. Such criterion is quite robust because the quadtree contrast-coding information is brightness independent (block mean values are removed). The only parameter in such process is the expected stripe size. In fact, the stripe width and height do not exhibit high variance values, as the eventual stripe size mostly depends on sensor positioning. Thus training a stripe extractor just consists in averaging the stripe size from examples containing license plates.



(b)

Fig. 3 – *(a)* Sample input picture and its quadree decomposition; *(b)* candidate stripes extracted from (a) and their associate aspect scores.

	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

Table I. Scores assigned to candidate stripes. Shaded row: the final selection

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 before undergoing VQ-based location.

2.5 Stripe scoring and final license plate location

After stripe extraction has worked out a set of candidate regions, the location module first compiles the set of codewords used to encode the blocks in the extracted stripes. Summing up the related probabilities 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 codeword scores takes into account the actual content of the coded blocks. The location process eventually selects the highest-score stripe from the final sorted list. Figure 2 presents a schematic representation of the complete system, and indirectly points out the deep-rooted link with the VQ-based compression paradigm.

The schema need not issue a location result for each input image, as the system can "reject" an input signal and prompt a null output. This useful feature is easily attained: if no candidate stripe exhibits a satisfactory score, no image region meets the reliability requirement for license-plate location. The rejection ability is actually boosted by using VQ to inspect a stripe's content.

3. EXPERIMENTAL RESULTS

The target ATS supported vehicle identification in a parking area in Madrid, Spain. The location system processed input fields (i.e., interlaced half frames) from a standard PAL camera; the input signal was converted into a digital picture holding 768x256 pixels with 256 gray-scale levels (8 bpp). The output of the location process was fed into a dedicated OCR module for license-plate interpretation [7]. In the current PC-based C++ implementation, a complete image-understanding cycle took about 200 ms on a standard Intel Pentium board (200Mhz). This performance proved cost-effective for the target application. The system training involved ten images acquired under different environmental conditions.

The use of VQ coding for archival purposes is motivated by the superior performance of that method at high compression ratios (i.e., Cr ≥ 40), as compared with standard JPEG algorithm [6].

3.1 Sample of the system operation

Figure 3a displays a sample of system input and its quadtree representation; Figure 3b presents the five extracted candidates; finally, Table I gives the scores associated with each stripe and points out the correct license-plate location. Anyway, 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 specific OCR adopted, a set of at most two stripes give a valid OCR input.

3.2 Location performance

The test set used for validating the method included more than 300 pictures, taken under quite different environmental conditions For the OCR module adopted, a location process is considered successful if the license plate appears in the first two stripes of the list. Results in Table II indicate that the overall system performance is satisfactory, as it exhibits a 2% error rate. This performance meets industrial requirements, and compares favourably with related approaches in the literature.

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

Table II. Statistical summary of location results

5. REFERENCES

[1] Comelli P, Ferragina P, Granieri MN, Stabile F "Optical recognition of motor vehicle license plates" *IEEE Trans. on Vehicular Technology*, 1995, vol.44, No.4, pp.790-799

[2] Postolache A, Trecat J "Pyramidal approach to license plate segmentation" *J. Electronic Imaging*, 1996, vol.5, No.3, pp.402-409

[3] Draghici S "A neural network based artificial vision system for license plate recognition" *Int.J. of Neural Systems*, 1997, vol.8, No.1, pp.113-126

[4] Gray RM "Vector Quantization" IEEE ASSP Mag., 1984, vol.1, No.2, pp.4-29

[5] Nasrabadi N, King R "Image Coding Using Vector Quantization: a Review" *IEEE Trans. on Communications*, pp. 957-971, Aug. 1988

[6] Ridella S, Rovetta S, Zunino R "Plastic algorithm for adaptive vector quantization" *Neural Computing and Applications*, 1998, vol.7, No.1, in press

[7] Ancona F, Colla AM, Rovetta S, Zunino R "Implementing probabilistic neural networks" *Neural Computing and Applications*, 1997, vol:5, pp. 152-159