A License Plate-Recognition Algorithm for Intelligent Transportation System Applications

Christos Nikolaos E. Anagnostopoulos, *Member, IEEE*, Ioannis E. Anagnostopoulos, *Member, IEEE*, Vassili Loumos, *Member, IEEE*, and Eleftherios Kayafas, *Member, IEEE*

Abstract—In this paper, a new algorithm for vehicle license plate identification is proposed, on the basis of a novel adaptive image segmentation technique (sliding concentric windows) and connected component analysis in conjunction with a character recognition neural network. The algorithm was tested with 1334 natural-scene gray-level vehicle images of different backgrounds and ambient illumination. The camera focused in the plate, while the angle of view and the distance from the vehicle varied according to the experimental setup. The license plates properly segmented were 1287 over 1334 input images (96.5%). The optical character recognition system is a two-layer probabilistic neural network (PNN) with topology 108-180-36, whose performance for entire plate recognition reached 89.1%. The PNN is trained to identify alphanumeric characters from car license plates based on data obtained from algorithmic image processing. Combining the above two rates, the overall rate of success for the license-plate-recognition algorithm is 86.0%. A review in the related literature presented in this paper reveals that better performance (90% up to 95%) has been reported, when limitations in distance, angle of view, illumination conditions are set, and background complexity is low.

Index Terms—Image processing, license plate recognition (LPR), optical character recognition (OCR), probabilistic neural network (PNN).

I. INTRODUCTION

DURING the past few years, intelligent transportation systems (ITSs) have had a wide impact in people's life as their scope is to improve transportation safety and mobility and to enhance productivity through the use of advanced technologies. ITSs are made up of 16 types of technology-based systems. These systems are divided into intelligent infrastructure systems and intelligent vehicle systems [1]. In this paper, a computer vision and character recognition algorithm for a license plate recognition (LPR) is presented to be used as a core for intelligent infrastructure like electronic payment systems (toll payment, parking fee payment), freeway, and arterial management systems for traffic surveillance. Moreover, as increased security awareness has made the need for vehicle-based authentication technologies extremely significant, the proposed system may be employed as access control system for monitoring of unauthorized vehicles entering private areas.

The license plate remains as the principal vehicle identifier despite the fact that it can be deliberately altered in fraud situations or replaced (e.g., with a stolen plate). Therefore, ITSs rely heavily on robust LPR systems. The focus of this paper is on the integration of a novel segmentation technique implemented in an LPR system able to cope with outdoor conditions if parameterized properly. Specifically, the novel contributions are

1) a novel segmentation technique named sliding concentric windows (SCWs) used for faster detection of regions of interest (RoI); and
2) an algorithmic sequence able to cope with plates of various sizes and positions. More than one license plate can be segmented in the same image.

Additionally, with the above contributions, a trainable optical character recognition (OCR) system based on neural-network technology is presented, featuring the advantage of continuous improvement. The neural-network architecture is based on a previous original work of ours [2].

The paper is organized as follows. The next section constitutes a review of similar researches that have been reported in the literature. In Section III, the SCWs segmentation method is described, followed by the complete corpus of the proposed LPR algorithm. Experimental results and sample set formation are presented in Section V. Problems occurred as well as restrictions of the complete setup are discussed in detail in Section VI. In Section VII, a comparison between LPR systems already proposed in the literature is given, and finally, conclusions and future extensions are presented in Section VIII.

II. REVIEW OF OTHER TECHNIQUES

Recognition algorithms reported in previous research are generally composed of several processing steps, such as extraction of a license plate region, segmentation of characters from the plate, and recognition of each character. Papers that follow this three-step framework are covered according to their major contribution in this section. The major goal of this section is to provide a brief reference source for the researchers involved in license plate identification and recognition, regardless of particular application areas (i.e., billing, traffic surveillance etc.).

A. License Plate Detection

As far as extraction of the plate region is concerned, techniques based upon combinations of edge statistics and mathematical morphology [3]–[6] featured very good results. In these methods, gradient magnitude and their local variance in an image are computed. They are based on the property that the brightness change in the license plate region is more remarkable and more frequent than otherwise. Block-based processing is also supported [7]. Then, regions with a high edge magnitude and high edge variance are identified as possible license plate regions. Since this method does not depend on the edge of license plate boundary, it can be applied to an image with unclear license plate boundary and can be implemented simply and fast. A disadvantage is that edge-based methods alone can hardly be applied to complex images, since they are too sensitive to unwanted edges, which may also show high edge magnitude or variance (e.g., the radiator region in the front view of the vehicle). In spite of this, when combined with morphological steps that eliminate unwanted edges in the processed images, the license plate extraction rate is relatively high and fast compared to other methods, as shown in Table VI.

Color or gray-scale-based processing methods are proposed in the literature for license plate location [8]–[12]. Crucial to the success of the color (or gray level)-based method is the color (gray level) segmentation stage. On the other hand, solutions currently available do not provide a high degree of accuracy in a natural scene as color is not stable when the lighting conditions change. Since these methods are generally color based, they fail at detecting various license plates with varying colors. Though color processing shows better performance, it still has difficulties recognizing a car image if the image has many similar parts of color values to a plate region. An enhanced color texture-based method for detecting license plates (LPs) in images was presented in [13]. The system analyzes the color and textural properties of LPs in images using a support vector machine (SVM) and locates their bounding boxes by applying a continuous adaptive mean shift (CAMShift) algorithm. The combination of CAMShift and SVMs produced efficient LP detection as time-consuming color texture analysis for less relevant pixels were restricted, leaving only a small part of the input image to be analyzed. Yet, the proposed method still encountered problems when the image was extremely blurred or quite complex in color. An example of time-consuming texture analysis is presented in [14], where a combination of a "kd-tree" data structure and an "approximate nearest neighbor" was adopted. The computational resource demand of this segmentation technique was the main drawback, since it yielded an execution time unacceptable for LPR (34 s).

In [15], a method is developed to scan a vehicle image with N row distance and count the existent edges. If the number of the edges is greater than a threshold value, this manifests the presence of a plate. If in the first scanning process the plate is not found, then the algorithm is repeated, reducing the threshold for counting edges. The method features very fast execution times as it scans some rows of the image. Nonetheless, this method is extremely simple to locate license plates in several scenarios, and moreover, it is not size or distance independent.

Fuzzy logic has been applied to the problem of locating license plates [16]–[18]. The authors made some intuitive rules to describe the license plate and gave some membership functions for the fuzzy sets "bright," "dark," "bright and dark sequence," "texture," and "yellowness" to get the horizontal and vertical plate positions, but these methods are sensitive to the license plate color and brightness and need longer processing time from the conventional color-based methods. Consequently, in spite of achieving better results, they still carry the disadvantages of the color-based schemes.

Gabor filters have been one of the major tools for texture analysis. This technique has the advantage of analyzing texture in an unlimited number of directions and scales. A method for license plate location based on the Gabor transform is presented in [19]. The results were encouraging (98% for LP detection) when applied to digital images acquired strictly in a fixed and specific angle, but the method is computationally expensive and slow for images with large analysis. For a two-dimensional (2-D) input image of size $N \times N$ and a 2-D Gabor filter of size $W \times W$, the computational complexity of 2-D Gabor filtering is in the order of $W^2 N^2$, given that the image orientation is fixed at a specific angle. Therefore, this method was tested on small sample images and it was reported that further work remain to be done in order to alleviate the limitations of 2-D Gabor filtering.

Genetic programming (GP) [20] and genetic algorithms (GAs) [21], [22] were also implemented for the task of license plate location. GP is usually much more computationally intensive than the GAs, although the two evolutionary paradigms share the same basic algorithm. The higher requirements in terms of computing resources with respect to the GAs are essentially due to the much wider search space and to the higher complexity of the decoding process as well as of the crossover and mutation operators. The authors indicate that the research carried out in [20]–[22], despite being encouraging, is still very preliminary and requires a deeper analysis. While the authors in [20] and [22] did not report clearly the results of their work, in [21], the identification ratio reached 80.6% on average, with a very fast execution time (0.18 s). In [22], the GA was presented for license plate location in a video sequence.

In the method using Hough transform (HT), edges in the input image are detected first. Then, HT is applied to detect the license plate regions. In [23], the authors acknowledge that the execution time of the HT requires too much computation when applied to a binary image with great number of pixels. As a result, the algorithm they used was a combination of the HT and a Contour algorithm, which produced higher accuracy and faster speed so that it can be applied to real-time systems. However, as HT is very sensitive to deformation of boundaries this approach has difficulty in extracting license plate region when the boundary of the license plate is not clear or distorted or the images contain lots of vertical and horizontal edges around the radiator grilles. This method achieved very good results when applied to close shots of the vehicle.

In [24], a strong classifier was trained for license plate identification using the adaptive boosting (AdaBoost) algorithm. When executed over several rounds, AdaBoost selects the best performing weak classifier from a set of weak classifiers, each

acting on a single feature, and, once trained, combines their respective votes in a weighted manner. This strong classifier is then applied to subregions of an image being scanned for likely license plate locations. Despite the fact that this paper has shown to be promising for the task of license plate detection, more work needs to be done. Additionally, since the classifier is applied to subregions of specific dimension, the system could not detect plates of different size or images acquired from different view/distance without retraining.

A wavelet transform-based method is used in [25] for the extraction of important contrast features used as guides to search for desired license plates. The major advantage of wavelet transform, when applied for license plate location, is the fact that it can locate multiple plates with different orientations in one image. Nevertheless, the method is unreliable when the distance between the vehicle and the acquisition camera is either too far or too close.

Symmetry is also used as a feature for car license plate extraction. The generalized symmetry transform (GST) produces continuous features of symmetry between two points by combining locality constraint and reflectional symmetry. This process is usually time consuming because the number of possible symmetrical pixels in the image is huge. Moreover, a rotated or perspectively distorted car license plate image is impossible to detect. In [26], the authors propose a scan line decomposition method of calculating GST in order to achieve considerable reduction of computational load. The result is indeed encouraging as far as the computational time is concerned, but since the scan line-based GST evaluates symmetry between a pair of edge pixels along the scan lines, the execution time increases linearly with respect to the radius of the searching area. Thus, the algorithm set limits to its effective distance, as a closer view of the plate results to increased processing time. Moreover, this approach is insufficient when rotated or distorted plates appear.

In addition, various neural-network architectures [27]–[29] are proposed and implemented for plate identification, namely the pulse coupled neural networks (PCNNs), the time delay neural networks (TDNNs), and the discrete time cellular neural networks (DTCNNs). In [27], it was demonstrated that the computationally most intensive steps in LPR could be realized by DTCNNs. The tests demonstrated that the DTCNNs were capable of correctly identifying 85% of all license plates. The total system contained several DTCNNs whose templates were constructed by combining the appropriate morphological operations and traditional filter techniques. This paper was an extension of a previous work of the authors [17], where they used fuzzy logic rules for license plate location.

In [28], the PCNN is used to generate candidate regions that may contain a license car plate. Candidate regions are generated from pulsed images, output from the PCNN. The results of these applications indicate that the PCNN is a good preprocessing element. The results have shown than in spite of being encouraging (85% for plate detection), a lot of research still remains to be performed.

Impressively good results were achieved in [29], where the TDNN schema was implemented. TDNN is a multilayer feedforward network whose hidden neurons and output neurons are replicated across a time. It has the ability to represent relationships between events in time and to use them in trading relations for making optimal decisions. In this paper, the license plate segmentation module extracts the license plate using two TDNNs as filters for analyzing color and texture properties of the license plate. The results were remarkable in terms of speed and accuracy with the drawback of computational complexity, but as the method is color based, it is also country specific. It should be also noted that as the system was designed to check a fixed region, there was a rough knowledge of possible plate location, which also explains the fast execution time versus the complexity of the algorithm.

In [30], a method based on vector quantization (VQ) to process vehicle images is presented. This technique makes it possible to perform superior picture compression for archival purposes and to support effective location at the same time. Compared to classical approaches, VQ encoding can give some hints about the contents of image regions. Such additional information can be exploited to boost location performance.

A sensing system with a wide dynamic range has been manufactured in [31] to acquire fine images of vehicles under varied illumination conditions. The developed sensing system can expand the dynamic range of the image (almost double in comparison with conventional CCDs) by combining a pair of images taken under different exposure conditions. This was achieved as a prism beam splitter installed a multilayered filter, and two charge-coupled devices are utilized to capture those images simultaneously.

B. Character Segmentation

The license plate candidates determined in the previous stage are examined in the license number identification phase. There are two major tasks involved in the identification phase: character segmentation and character recognition. A number of techniques to segment each character after localizing the plate in the image have also been developed, such as feature vector extraction and mathematical morphology [32], and Markov random fields (MRFs) [22].

The work in [32] proposed a novel adaptive approach for character segmentation and feature vector extraction from seriously degraded images. An algorithm based on the histogram automatically detects fragments and merges these fragments before segmenting the fragmented characters. A morphological thickening algorithm automatically locates reference lines for separating the overlapped characters. A morphological thinning algorithm and the segmentation cost calculation automatically determine the baseline for segmenting the connected characters. Basically, this approach can detect fragmented, overlapping, or connected characters and adaptively apply one of three algorithms without manual fine tuning. The results are very promising and encouraging, indicating that the method could be used for character segmentation in plates with not easily distinguishable characters during off-line operation, but since the algorithm is computationally complex, it cannot be proposed for real-time LPR.

Different from many existing single-frame approaches, the method proposed in [22] simultaneously utilizes spatial and

temporal information. First, it models the extraction of characters as a MRF, where the randomness is used to describe the uncertainty in pixel label assignment. With the MRF modeling, the extraction of characters is formulated as the problem of maximizing *a posteriori* probability based on a given prior knowledge and observations. Later on, the GA with a local greedy mutation operator is employed to optimize the objective function. This method was developed for license plate segmentation in video sequences but the segmentation results were far from suitable for automatic character recognition.

The brightness distribution of various positions in a license plate image may vary because of the condition of the plate and the effect of lighting environment. Since the binarization with one global threshold cannot always produce useful results in such cases, adaptive local binarization methods are used [31], [33]. In many local binarization methods, an image is divided into $m \times n$ blocks, and a threshold is chosen for each block. A similar method, proposed by Sauvola, was adopted in our LPR system for character segmentation and is discussed later on.

C. Character Recognition

For the recognition of segmented characters, numerous algorithms exploited mainly in optical character-recognition applications utilized hidden Markov models (HMMs) [23], [34], neural networks [2], [15], [17], [18], [35], [36], Hausdorff distance [3], SVM-based character recognizer [29], and template matching [11], [37].

When HMMs are employed, the recognition begins with a preprocessing and a parameterization of the RoIs detected in the previous phase. The recognition result in [34] was reported to be 95.7% after a complex procedure of preprocessing and parameterization for the HMMs. The authors also reported that the width of the plate in the image after rescaling lies between 25% and 75% of the width of the image (between 200 and 600 pixels). This reveals the necessity for good character analysis when implementing HMM, which posses a restriction in the effective distance of the recognition system. This prerequisite is also featured in [23], where the recognition results reached 92.5%.

Multilayer perceptron neural networks were used for license plate character identification [15], [17]. The training method for this kind of network is the error backpropagation (BP). The network has to be trained for many training cycles in order to reach a good performance. This process is rather time consuming, since it is not certain that the network will learn the training sample successfully. Moreover, the number of the hidden layers as well as the respective neurons have to be defined after a trial and error procedure. In [17], an MLP containing 24 input, 15 hidden, and 36 output neurons was trained to recognize 36 characters of the Latin alphabet (26 letters + 10 numerals). The input neurons were fed by 24 features generated previously from a DTCNN. The network was applied to segmented license plates for character recognition, achieving excellent results (98.5%) in a large set of images (10 000). Nonstandard feed forward NN were also tested, such as the enhanced neural network [38].

In [18], a self-organized neural network based on Kohonen's self-organized feature maps (SOFMs) was implemented

to tolerate noisy, deformed, broken, or incomplete characters acquired from license plates, which were bent and/or tilted with respect to the camera. The method focuses on accuracy at the cost of increased complexity and execution speed. The success rate for character identification in a large set of 1061 license plates in various viewpoints (combinations of angle and distance) was a remarkable 95.6%.

Probabilistic neural networks (PNNs) for LPR was first introduced in [2]. In this paper, we trained and tested successfully two probabilistic networks: one for alphabet recognition and the other for number recognition. In a small testing set (280 characters), the performance ratio reached 97.5% and 95.0%, respectively. PNNs are the neural-network implementation of kernel discriminate analysis and were introduced into the neural-network literature by Specht [39]. More neurons may be required than standard feedforward BP networks, but PNN can be designed in a fraction of the time it takes to train standard feed-forward networks. This useful feature comes with the drawbacks of larger memory requirements and slightly slower execution speed compared to conventional neural networks [40]. The recognition rates that are lately reported in the literature are very encouraging when PNNs were trained and tested in noisy, tilted, and degraded patterns [2], [35], [36]. In [36], the authors reported an impressive recognition rate that reaches 99.5%.

Hausdorff distance is a method to compare two binary images (or two sets of "active" pixels). Hausdorff distance has all the mathematical properties of a metric. Its main problem is the computational burden. Its recognition rate is very similar to that obtained with neural-network classifiers, but it is slower. Therefore, it is good as a complementary method if real-time requirements are not very strict. It can be used if there is extra time to improve results from another classifier [3]. Additionally, in [29], the authors designed a system implementing four SVMs and report an impressive average character recognition rate of 97.2%. The architecture, however, is strictly designed for Korean plates, thus leaving no response in license plates of other countries.

A suitable technique for the recognition of single font (not rotated) and fixed size characters is the pattern matching technique. Although this one is preferably utilized with binary images, properly built templates also obtained very good results for grey level images. A similar application is described in [11], where the authors used a normalized cross correlation operator. The recognition process was based on the computation of the normalized cross correlation values for all the shifts of each character template over the subimage containing the license plate. It was reported that more than 90% of the CPU time was spent for the computation of the cross correlation measures between the various templates and the relative subimage. However, as the subimage is a small image, the problem of computational time is overcome. The time spent to run the complete system was about 1.1 s per image. Therefore, unlike the license plate-location problem, template-matching techniques can be used for character recognition in plates, provided that they are fixed size and not tilted or rotated. Template matching is also implemented successfully in [37].
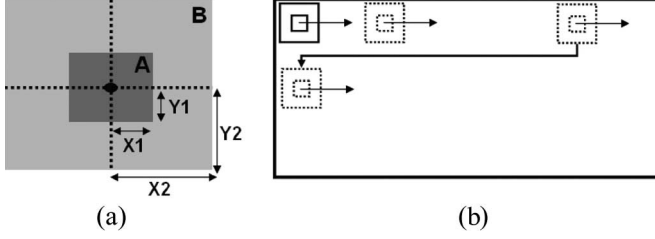
Fig. 1. (a) Concentric windows and (b) scanning the image.

In this paper, an algorithm implementing a novel adaptive image segmentation technique (SCWs) and connected component analysis is considered for license plate location and character segmentation. For the OCR task, a PNN is trained to identify alphanumeric characters, which were previously isolated from the candidate area.

III. SCW SEGMENTATION METHOD

Hypothetically, license plates can be viewed as irregularities in the texture of the image and therefore abrupt changes in the local characteristics of the image, manifest probably the presence of a license plate. Based on the above, this paper proposes a novel adaptive segmentation technique named SCWs. The method is developed in order to describe the "local" irregularity in the image using image statistics such as standard deviation and mean value. The algorithm was developed implementing the following steps.

1) Two concentric windows A and B of size $(2X_1) \times (2Y_1)$ pixels and $(2X_2) \times (2Y_2)$, respectively, were created for the first pixel of the image (upper left corner). The windows are presented in Fig. 1(a).
2) Statistical measurements in windows A and B were calculated.

Definition of a segmentation rule: If the ratio of the statistical measurements in the two windows exceeds a threshold set by the user, then the central pixel of the windows is considered to belong to an RoI.

Therefore, let x, y be the coordinates of the examined pixel in inspected image I. The pixel value in the respective coordinates x, y of the resulting image I_{AND} is set either 0 (no RoI) or 1 (RoI) according to

$$in\ I_1(x,y) \Rightarrow \begin{cases} I_{\mathrm{AND}}(x,y)=0, & \text{if } \frac{M_B}{M_A} \leq T \\ I_{\mathrm{AND}}(x,y)=1, & \text{if } \frac{M_B}{M_A} > T \end{cases} \quad (1)$$

where M is the statistical measurement (mean value or standard deviation). The two windows slide until the entire image is scanned, as shown in Fig. 1(b). The parameters X_1, X_2, Y_1, Y_2, and T could be updated according to the specific application. However, after experimentation in the original SCW method, it was interestingly found that if the ratio of the concentric windows is near the ratio of an object to be segmented, then the segmentation results will be optimal. Therefore, the parameters X_1, X_2, Y_1, and Y_2 were selected on the basis of the above observation. Specific details are presented in the respective algorithmic steps that follow in Sections IV-A and B.

TABLE I
PLATE SEGMENTATION PSEUDOCODE

Input: image I

1 **for** each pixel in image I_1,
  run SCW method in I_1
  {parameters of SCW: $X_1, Y_1, X_2, Y_2,$ T and M=mean value; }
  //parameters are set according the application setup. Details can be found in Table IV.
    name the resulting image I_{AND}
2   **perform** image masking

    calculate image I_2, where $I_2 = I_1 \cap I_{AND}$;
3   **for** each pixel in image I_2,
    { **binarize** I_2 using Sauvola method with parameters: k=0.5, R=128, b=10; }
    name the resulting image I_3
4   **perform** binary measurements in I_3 objects
    { **retrieve** objects which fulfill the following measurements:
      (Aspect Ratio>2) **and**
      (Orientation<35 degrees) **and**
      (Euler Number > 3) };
5   **count** n, where n: number of objects detected
    if n>0 in I_3,
    { $I_3 = I_4$ and proceed to step 6;}
    if n=0 in I_3, {
    invert I_1 ($I_1 = -I_1$) and **perform** steps 1,2,3, 4 and 5 again;
    if n=0 in the inverted image I_1
    {print "No license plates found" }
      else { $I_3 = I_4$ and proceed to step 6;}
6   **for** i=1:n in image I_4
    { **detect** coordinates of Xmin Xmax Ymin Ymax for object i;}
    Output: matrix A(i) where: A(i)= (Xmin,Xmax,Ymin,Ymax);

On the other hand, there was no evidence that there is a best possible way to choose a value for threshold T, and therefore, this is to be decided according the application after trial-and-error procedure.

IV. LICENSE PLATE RECOGNITION

The LPR sequence, which is proposed in this paper, consists of two distinct parts. The first one deals with the detection of the RoI, i.e., the license plate. The second part includes operations for the successful segmentation of the license plate characters along with an artificial neural network, which performs the OCR task.

A. First Part: License Plate Segmentation

This part consists of preprocessing, in order to detect the RoI in ambient illumination conditions. The preprocessing component consists of four tasks: implementation of the SCW segmentation method, image masking, binarization with Sauvola method, and finally, connected component labeling and binary measurements, which are arranged in sequence. The license plate-segmentation pseudocode is indicated in Table I, and the respective flowchart is highlighted in Fig. 2.

The input to this module is a gray-level image. The resulting image after the SCW method is the binary image I_{AND}, which is the mask to be used in the AND masking operation. The parameters X and Y for the concentric windows were selected according the aspect ratio of the objects of interest, which are the license plate regions in this step. As license plates in Greece have an aspect ratio (height/width) near 1/3
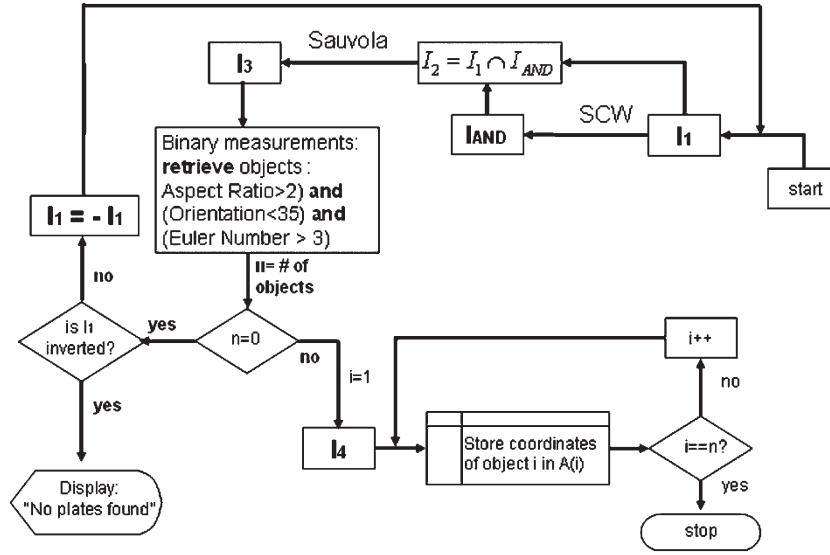
Fig. 2. Flowchart indicating the license plate location algorithm.

(or width/height ≈ 3), the Xs in the concentric windows were selected to be three times bigger than the Ys (e.g., $X_1 = 12$, $Y_1 = 4$, $X_1 = 6$, $Y_1 = 2$). In addition, threshold T was set after a trial-and-error evaluation. The specific values can be found in Table IV in Section V, where the complete sample set is described.

Image I_2 is the outcome from the logical AND operation between images I_{AND} and the original image I_1. For the binarization of image I_2, the locally adaptive thresholding method of Sauvola [41] was chosen. The adaptive thresholding techniques are characterized by the calculation of a threshold value at each pixel. The value of the threshold depends upon some local statistics like range, variance, and surface fitting parameters or their logical combinations. It is typical of locally adaptive methods to have several adjustable parameters. The threshold $T(x, y)$ will be indicated as a function of the coordinates x, y. This method adapts the threshold according to the local mean and standard deviation over a window size of $b \times b$. The threshold at pixel (x, y) is calculated as

$$T(x, y) = m(x, y) + \left[1 + k.\left(\frac{\sigma(x, y)}{R} - 1\right)\right] \quad (2)$$

where $m(i, j)$ and $\sigma(i, j)$ are the local sample mean and variance, respectively. Sauvola suggests the values of $k = 0.5$, $R = 128$, and $b = 10$, which were adopted in this algorithm. Thus, the contribution of the standard deviation becomes adaptive. For example, in the case of badly illuminated areas, the threshold is lowered. Hence, image I_2 is binarized using

$$I_3(x, y) = \begin{cases} 1, & \text{if } I_2(x, y) \geq T(x, y) \\ 0, & \text{if } I_2(x, y) < T(x, y) \end{cases}. \quad (3)$$

Connected components analysis (CCA) is a well-known technique in image processing that scans an image and labels its pixels into components based on pixel connectivity (i.e., all pixels in a connected component share similar pixel intensity values) and are in some way connected with each other (either four-connected or eight-connected). Once all groups have been determined, each pixel is labeled with a value according to the component to which it was assigned.

CCA works on binary or gray-level images and different measures of connectivity are possible. For the present application, we apply CCA in binary images searching in eight-connectivity. Extracting and labeling of various disjoint and connected components in an image is central to many automated image analysis applications as many helpful measurements and features in binary objects may be extracted, such as area, orientation, moments, perimeter aspect ratio, just to name a few of them [42].

Following the successful connected component labeling in image I_3, measurements such as the orientation, the aspect ratio, and the Euler number for every binary object in the image are performed. Those objects whose measurements fulfill the criteria orientation $< 35°$, $2 < $ Aspect ratio < 6, and Euler number > 3 are considered as candidate plate regions, and their coordinates are stored in matrix A.

The aspect ratio (also called elongation or eccentricity) can be found by scanning the image and finding the minimum and maximum values on the row and columns, where the object lies [42]. This ratio is defined by

$$\frac{c_{\text{MAX}} - c_{\text{MIN}} + 1}{r_{\text{MAX}} - r_{\text{MIN}} + 1}.$$

where c indicates column, and r indicates row. The axis of least second moments [42] provides information about the orientation of the object. It is defined as follows:

$$\tan(2\theta_i) = 2\frac{\sum_{r=0}^{N-1}\sum_{c=0}^{N-1} rcI_i(r, c)}{\sum_{r=0}^{N-1}\sum_{c=0}^{N-1} r^2 I_i(r, c) - \sum_{r=0}^{N-1}\sum_{c=0}^{N-1} c^2 I_i(r, c)}.$$

The Euler number [42] of an image is defined as the number of objects minus the number of holes. For a single object, it provides the number of the closed curves that the object contains. It is obvious that in the binary image, the characters of
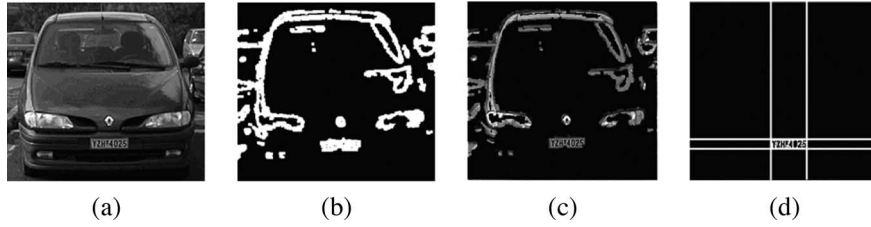
Fig. 3. Steps for license plate segmentation. (a) Initial image. (b) Result of SW segmentation technique after the application of the segmentation rule and thresholding. (c) Image I_2 after the image masking step. (d) Plate detection.
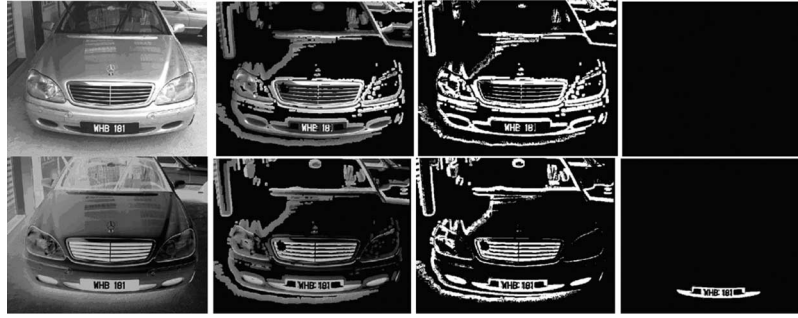

Fig. 4. Steps for license plate segmentation with black background and white characters. First row: License plate not found as there is no object found that fulfills the Euler number criterion. Second row: The initial image was inverted, and the license plate was located.

the license plate represent "holes" in the object "plate" Using this binary measurement, the objects that include more than three holes in them are preserved. The Euler number value was set $E > 3$, in order to enable the algorithm to detect plates containing no less than four characters. Fig. 3 portrays a sequence of successful license plate identification.

However, the above procedure does not guarantee that license plates with dark background and white characters will be successfully detected. Therefore, as indicated in Fig. 2 (or in step 5 in the pseudocode), if $n = 0$, where n is the number of candidate objects, I_1 is inverted, and steps 1, 2, 3, and 4 are once again executed in the inverted image. If again $n = 0$ in step 5, then there is no license plate in the image (or the algorithm fails to find it), and therefore, the message "No license plates found" appears. Conversely, if $n > 0$ (at least one object exists), then $I_3 = I_4$, and I_4 is forwarded to step 6. Preprocessing repeats until all candidate objects' coordinates are stored in matrix A. Fig. 4 highlights an example of successful license plate identification with dark background.

B. Second Part (License Plate Processing)

This part consists of preprocessing, in order to isolate the characters in the license plate image and then forward them in the neural network for identification. Table II presents the pseudocode of the license plate processing steps, while the flowchart is shown in Fig. 5.

Considering a candidate region, the coordinates are recalled from matrix A in order to crop the respective subimage. This image is transformed to a standard size of 75×228 pixels using the bicubic interpolation method and then subjected to the SCW segmentation method with the following parameters: $X_1 = 2$, $Y_1 = 5$, $X_2 = 4$, $Y_2 = 10$, and $T = 0.7$, where the measurement is the standard deviation value. As indicated in Section III, the above parameters are selected according the aspect ratio

TABLE II
PLATE PROCESSING AND CHARACTER RECOGNITION

for j=1:n, where n is the number of detected plates
{
Input: matrix A(j)
1 **crop** I1 using the coordinates in matrix A(j)
name the resulting image I$_5$;
2 **resize** I$_5$ to 75x228 pixels using bicubic interpolation;
3 **for** each pixel in image I$_5$,   {
run SCW method in I$_5$;
parameters: X$_1$=2,Y$_1$=5, X$_2$=4, Y$_2$=10, T=0.7
and M=standard deviation value;   }
name the resulting image I$_6$;
4 **inverse** image I$_6$;
name the resulting image I$_7$;
5 **perform** binary measurements in I$_7$ objects
{ **retrieve** objects which fulfill the following measurements:
(Orientation>75 degrees) **and**
(Height>32 pixels) };
name the resulting image I$_8$;
6 **identify** bounding box of binary objects(characters) in I$_8$
using column and row standard deviation criterion;
7 **perform** character segmentation in I$_8$;
8 **count** c, where c: number of characters detected in I$_8$
9 **for** a=1:c ,
{ **transform** character to 9x12 pixels
create input vector [1 108]; }   // 9x12=108
10 **forward** input vector to PNN and get decision;
}

of the objects of interest for optimal results. In the specific task, the objects to be emphasized (the characters) have an aspect ratio (height/width) near to 5/2. Therefore, the parameters for the inner window was set $X_1 = 2$, $Y_1 = 5$ and for the outer window $X_2 = 4$, $Y_2 = 10$ in order to ensure an aspect ratio of 5/2 for both concentric windows. The threshold T was set to 0.7 after a trial and error for the optimization of the results.

Following the inversion of the resulting image and object labeling, the orientations and heights of connected components (objects) are then calculated. The components whose
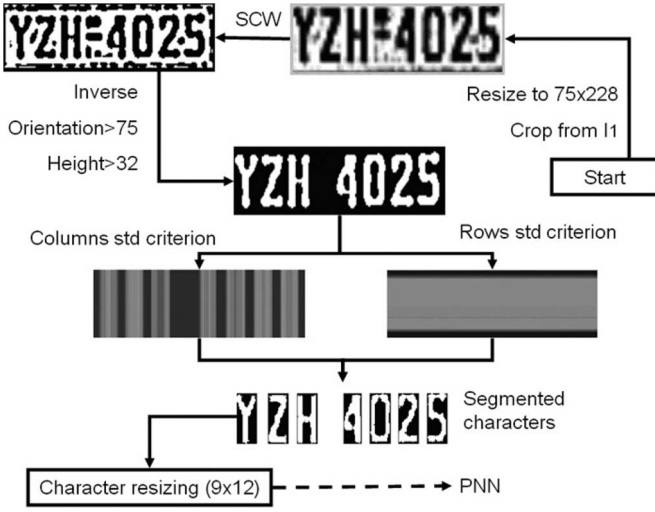
Fig. 5. Flowchart indicating the license plate processing and character recognition algorithm.

Fig. 6. Successful segmentation of characters following the SCW method. Original segmented plate (left), SCW result (middle), and character segmentation after the binary measurements (height, orientation). Parameters are $X_1 = 2$ $Y_1 = 5$, $X_2 = 4$, $Y_2 = 10$, and $T = 0.7$.

measurements do not fulfill specific rules (orientation $> 75°$ and height > 32 pixels; see Table II, step 5) are deleted. The remaining objects are then forwarded to PNN after a well-defined preparation process, which includes character segmentation calculating the standard deviation of columns and rows, and, finally, transformation of each character to the size of the input vector of the PNN. Fig. 6 presents examples of license plate processing with the proposed algorithm.

C. Third Part (Character Recognition)

A PNN uses a supervised training set to develop distribution functions within a pattern layer. These functions, in the recall mode, are used to estimate the likelihood of an input feature vector being part of a learned category or class. The learned patterns can also be combined or weighted, with the *a priori* probability of each category, to determine the most likely class for a given input vector. If the relative frequency of the categories is unknown, then all categories can be assumed to be equally possible and the determination of category is solely based on the closeness of the input feature vector to the distribution function of a class.

PNNs contain an input layer, with as many elements as there are separable parameters needed to describe the objects to be classified as well as a middle/pattern layer, which organizes the training set so that an individual processing element represents each input vector.

Finally, they have an output layer also called summation layer, which has as many processing elements as there are classes to be recognized. Each element in this layer is combined via processing elements within the pattern layer, which relate to the same class and prepares that category for output. A PNN is guaranteed to converge to a Bayesian classifier, provided that it is given enough training data [40].

The topology of the proposed PNN is 108-180-36 and it is presented in Fig. 7. The input layer consists of 108 nodes, which correspond to the 108-input vector ($108 = 9 \times 12$). The second layer is the middle/pattern layer, which organizes the training set in such a way that an individual processing element represents each normalized input vector. Therefore, it consists of 180 nodes that correspond to the total amount of the used training patterns (five image patterns of each one of the 36 characters). Finally, the network has an output layer consisting of 36 nodes, representing the 36 possible characters in the plate.

A conscience full competitive learning mechanism between the weights of the input and middle layer tracks how often the outputs win the competition with a view of equilibrating the winnings, implementing an additional level of competition among the elements to determine which processing element is going to be updated. Assuming that O is the number of outputs ($O = 36$), the weight update function for the winning output is defined in (4). In this equation, y_i is the ith output vector that measures the distance between the input and the output neurons' weight vectors, x_j is the jth input vector, and iw_{ij} is the connection weight that links the processing element j with the processing element i.

Finally, f_i corresponds to the output's frequency of winning where $0 \leq f_i \leq 1/O$, and b_i defines the respective bias vector created by the conscience mechanism. Factor γ corresponds to a fine tuning training parameter

$$U(y) = \max_i(y_i + b_i) = \max_i \left(\sqrt{\sum_{j=1}^{108} (x_j - iw_{ij})^2} + b_i \right) \quad (4)$$

where

i	$1, 2, \ldots, 180$;
b_i	$\gamma \cdot [O \cdot (\beta \cdot (1 - f_i))]$ i: winner;
b_i	$\gamma \cdot [O \cdot (\beta \cdot f_i)]$ I: otherwise;
β	$1/2\sigma^2$, where σ is the standard deviation.

Between layers, the activation of a synapse is given by the Euclidean distance metric and the function, which mimics the neuron's synaptic interconnections. The activation and the cost function are defined by (5) and (6), respectively:

$$m(x_j(t), iw_{ij}) = \sqrt{\sum_j (x_j(t) - iw_{ij})^2} \quad (5)$$

$$J(t) = \frac{1}{2} \sum_j (d_j(t) - m(x_j(t), iw_{ij}))^2 \quad (6)$$

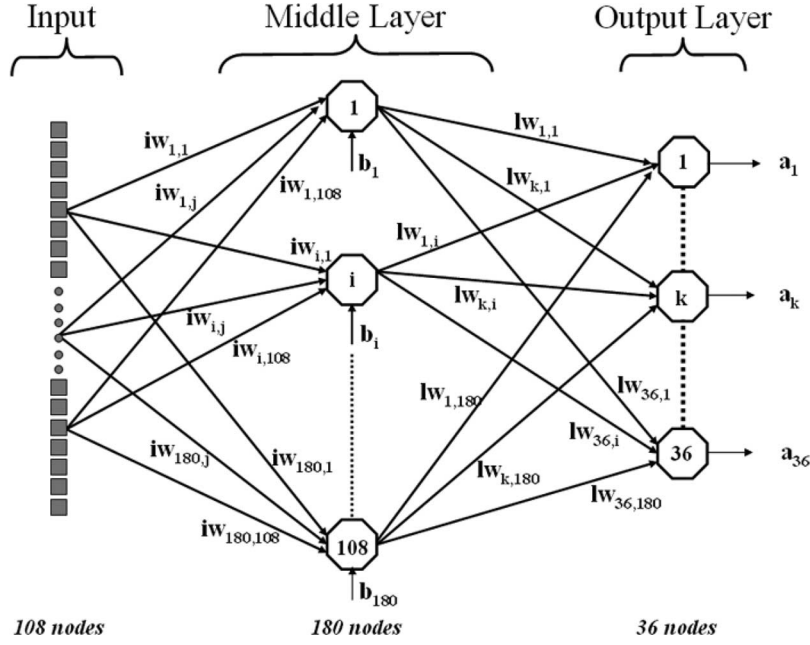$$\frac{\partial J(t)}{\partial m(x_j(t), iw_{ij})} = 0 \quad (7)$$

Fig. 7.   Architecture of the PNN.

where d_j is every desired response during the training epoch. It should be noted that toward cost function optimization in (7) should be satisfied.

The total number of synapses between the input and the pattern layer is 108×180, and they are presented in Fig. 7 as $iw_{i,j}$, where $1 \leq i \leq 180$, $1 \leq j \leq 108$, while between the pattern and the output layer, the total number is 180×36 and indicated as $lw_{k,l}$, where $1 \leq k \leq 36$, $1 \leq l \leq 180$.

In parallel, the proposed PNN uses a supervised training set to develop distribution functions within the middle layer. These functions are used to estimate the likelihood of the input being a learned character. The middle layer represents a neural implementation of a Bayes classifier, where the character class dependent probability density functions are approximated, using the Parzen estimator [43]. This estimator is generally expressed by $(1/n\sigma)\sum_{i=0}^{n-1} W((x - x_i)/\sigma)$.

Above, n is the sample size, x and x_i are the input and sample points, σ is the scaling parameter that controls the area's width considering the influence of the respective distance, and W is the weighting function [44]. This approach provides an optimum pattern classifier in terms of minimizing the expected risk of misclassifying an object. With the estimator, the approach gets closer to the true underlying class density functions, as the number of training samples increases, provided that the training set is an adequate representation of the class distinctions. The likelihood of an unknown character x which belongs to a given class is calculated according to

$$g_{i(c)} = \frac{1}{(2\pi)^{\frac{p}{2}} \sigma^p N_i} \sum_{j=0}^{(N_i-1)} e^{\frac{-(x-\bar{x}_{ij})^T(x-\bar{x}_{ij})}{2\sigma^2}} \tag{8}$$

where i reflects to the number of the class, j is the pattern layer unit, $\bar{x}_{ij}$ corresponds to the jth training vector from class i, and

c is the tested vector. In addition, N_i represents the respective training vectors for the ith class, p equals to the dimension of character x ($p = 108$), σ is the standard deviation, and $1/2\sigma^2$ outlines the beta (β) coefficient.

In other words, (8) defines the summation of multivariate spherical Gaussian functions centered at each of the training vectors $\bar{x}_{ij}$ for the ith class probability density function estimate.

Furthermore, in the middle layer, there is a processing element for each input vector of the training set and equal amounts of processing elements for each output class, in order to avoid one or more classes being skewed incorrectly. Each processing element in this layer is trained once to generate a high output value when an input matches the training vector. However, the training vectors do not need to have any special order in the training set, since the category of a particular vector is specified by the desired output.

The learning function simply selects the first untrained processing element in the correct output class and modifies its weights to match the training vector. The middle layer operates competitively, where only the highest match to an input prevails and generates an output.

The classifier was implemented in Visual C++ 6 and trained in a Pentium IV at 3.0 GHz with 512-MB RAM. The time needed for the completion of the training epoch was 12 s. It was evaluated that the neural network was not properly trained when $0.1 < \beta < 1$, due to large mean-square-error (mse) values. Further investigation over the influence of β in the learning ability of the classifier exposed to us that mse reduced when $3 < \beta < 10$. For $\beta = 0.01$ the mse was minimized. Therefore, during the training period, the "beta" coefficient for all the local approximators of the middle layer was set equal to 0.01. Additionally, training coefficient γ was set equal to 0.3 ($\gamma = 0.3$). A similar classifier was implemented in [45] for web page classification application.

Fig. 8.   Images from test sample 1 (frontal view) and test sample 2 (back view).

Fig. 9.   Images from test sample 3 (frontal view) and test sample 4 (back view).

Fig. 10.   Images from test sample 5. The one in the middle was captured using infrared illuminator and special camera for night operation.

V. SAMPLE SET—EXPERIMENTAL RESULTS

The complete testing image database consists of 1334 digital images from different five sample sets. The majority of the images represents Greek license plates from the natural scenes obtained in various illumination conditions. Sample sets 1 and 2 (Fig. 8) contain actual images acquired from a testbed set in the National Technical University of Athens (NTUA) campus, which consisted of a digital camera, a vehicle proximity sensor, the PC, and the communication cables. The images were not processed in real time but stored for later processing.

Since 1983, license plates in Greece have had a standard format of three letters and four numerals. The first one or two of the letters are indicative of the prefecture where the vehicle is registered. For instance, in the wide region of Athens (Attica), the first letter of the plate was originally set to "Y."

However, as the vehicle number increased, letter "Z" was also used as a first letter and very recently the letter "I." Until 2004, the front Greek license plate was a rectangular object of 33-cm width and 10-cm height, while the rear one was bigger (44 × 14 cm). From 2005, a new license plate format was established in Greece following the European standards. The new standard differs in size and not in content (e.g., there is still three letters and four numerals with the same prefecture coding). Due to the location of our test bed, the majority of the vehicles captured in our image sets were registered to the prefecture of Attica, where the first letter of the license plate is "Y" or "Z" or "I" followed by two other letters. As a result, the occurrence of those characters was higher compared to others. However, all the 26 alphabets (A to Z) appeared in satisfactory numbers in our samples, and therefore, the performance of the classifier was safely assessed. Fortunately, vehicles from other prefectures were captured (e.g., Fig. 11), as well as from other countries.

For sample sets 1 and 2, the camera used was a SONY XCD/X700 IEEE-1394 digital camera, with a Canon PHF3.5 lens with focal length 3.5 mm, which was triggered by the sensor located at the roadside of the experimental site. The distance between the digital camera and the vehicle was set to 4.5 m at a height of 1.2 m from the ground. The camera focused in the expected plate region.

Sample sets 3 and 4 (Fig. 9) represent the natural scenes of nonmoving vehicles obtained manually from the campus parking area using a Nikon Coolpix 885 adjusted to acquire 1024 × 768 pixel images. The distance between the camera and the vehicle varied from 4 up to 6 m at a height of 1.6–1.8 m form the ground. Moreover, the testing database was enriched with test sample 5 (Fig. 10), which includes digital images containing vehicles from various application setups. The images were downloaded from [46], representing automobiles of other countries. Table III indicates the composition of the complete image database.

As far as overall performance calculation is concerned, this can be achieved by calculating the percentage of license plates that have been correctly identified by the machine and verified by a person supervising the test sample [47]. This percentage is figured using

$$A = (P \times I)\% \qquad (9)$$

where A is the total system accuracy, and P, I is the percentage of successful plate segmentation and successful interpretation of entire plate content (successful recognition of all characters in the plate), respectively. Table IV depicts the performance of the proposed system for every sample set and the SCW settings for license plate segmentation. As in sample set 5, some license plates (e.g. from Hong Kong) have ratio width/height ≈ 2; the parameters were set to $X_1 = 8$, $Y_1 = 4$, $X_1 = 4$, and $Y_1 = 2$ to ensure that those license plates would be properly segmented. Moreover, the threshold T was set either 0.5 for daylight or 0.8 for night scenes.

The image-processing algorithmic steps as well as the PNN were completely built in Microsoft's Visual C++ 6.0 as standalone executable programs. Fig. 11 demonstrates a screen capture of the LPR graphical user interface (GUI) in use. In addition, Table V reports the required computational time for every step of the proposed algorithm. The entries of the table are the average execution time for 100 runs of the algorithm using an internal millisecond timer.

VI. PROBLEMS IDENTIFIED—RESTRICTIONS OF THE SYSTEM

The major problems in the proposed algorithmic sequence revolve around the varying light levels encountered during a 24-h period and the effect those lighting changes have on the image being forwarded to the OCR program as well as due to the physical appearance of the plates.

TABLE III
IMAGE DATABASE COMPOSITION

Samples	Acquired from	Description	Distance (meters)	Average plate size (pixels)	Pan, Tilt	Images
Sample set 1	Testbed	Campus entrance, morning (10 am-2 pm), Moving vehicles, frontal view	~4.5	90x28	25° 15°	427
Sample set 2	Testbed	Campus exit, afternoon (3-5pm), Moving vehicles, back view	~4.5	102x32	25° 15°	258
Sample set 3	Manually	Campus parking, morning (10 am-2 pm), Non-moving vehicles, frontal & back view	3-6	Various from: 61x19 up to 153x48	0-40° 0-25°	310
Sample set 4	Manually	Campus parking, afternoon(3-5p.m.), Non-moving vehicles, frontal & back view	3-6	Various from: 61x19 up to 153x48	0-40° 0-25°	154
Sample set 5	www [46]	Various road-side traffic scenes	various	Various from: 138x29 up to 255x37	various	185
Total number of images						**1334**

TABLE IV
ALGORITHM PERFORMANCE

Images	SCW settings for license plate segmentation M is mean value	Performance in plate segmentation	Performance in entire plate content	Overall performance for each sample set
Sample set 1	X1=12, Y1=4, X1=6, Y1=2,T=0.5	419/427 (98.1%)	372/419 (88.8%)	372/427 (87.1%)
Sample set 2	X1=12, Y1=4, X1=6, Y1=2,T=0.8	238/258 (92.2%)	208/238 (87.4%)	208/258 (80.6%)
Sample set 3	X1=12, Y1=4, X1=6, Y1=2,T=0.5	303/310 (97.7%)	280/303 (92.4%)	280/310 (90.3%)
Sample set 4	X1=12, Y1=4, X1=6, Y1=2,T=0.8	147/154 (95.5%)	129/147 (87.8%)	129/154 (83.8%)
Sample set 5	X1=8, Y1=4, X1=4, Y1=2, T=0.5 (day),T=0.8 (night)	180/185 (97.3%) 1287/1334 **(96.5%)**	159/180 (88.3%) 1148/1287 **(89.1%)**	159/185 (85.9%)
Overall Performance (success in entire content/total images)		1148/1334 **(86.0%)**		

Fig. 11. Software GUI. The vehicle in this screenshot is registered in the prefecture of Argolida (APB).

TABLE V
REQUIRED PROCESSING TIME

Version	Average time (milliseconds)
License plate segmentation	111
License plate processing	37
Character recognition	128
TOTAL TIME	276

As shown in Table IV, the algorithm failed to identify 47 plates. Twenty (20) of them represent plates with not easily distinguishable characters, either due to the fact that the plates were damaged or due to their physical appearance (i.e., extremely dirty plates or with stickers and unofficial stamps attached on their surface). The remaining failure cases were due to sudden change of environmental—illumination condition or incorrect settings during the fine tuning of the experimental

Fig. 12. Algorithm failure in sample set 3 and 4.

setup. In such an outdoor environment, illumination not only changes slowly as daytime progresses but may change rapidly due to changing weather conditions or passing objects (i.e., clouds). Examples are given in Fig. 12.

On the other hand, as far as license plate processing is concerned, the entire character content in 139 plates was not correctly identified (Table IV). In those 139 cases, at least one character classification error occurred, and subsequently, the system failed to recognize the complete plate at all.

A common failure was the event of two individual characters to be processed as one as they appear connected in the digital image. Despite the fact that the license plate processing part
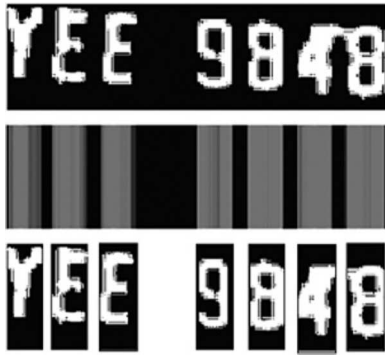
Fig. 13. Successful segmentation in connected characters.

has the ability to separate characters that are accidentally connected during the character segmentation process, 246 character segmentation errors were recorded. In Fig. 13, characters "4" and "8" are connected due to a screw located between them. However, in this case, the characters were separated successfully using the column and row standard deviation criterion (see Table II, line 6).

However, the majority of character misclassification problems was due to similar characters, such as 1 and I, 0 and O, 2 and Z, 5 and S, and 8 and B. Especially, misclassification occurred in the cases of "0"—"O" and "1"—"I." The evident solution is the introduction of *a priori* knowledge for the kind of character to be recognized (letter or number). However, with this assumption, the LPR process would be country specific.

It is evident that the restrictions of the proposed algorithm are strongly related to the environmental conditions during the acquisition of the database sample. As shown in Table III, the algorithm was not tested on a 24-h operation. However, sample 5 obtained from [46] includes 44 vehicle pictures (among others), which were obtained during night or late afternoon using infrared illuminators that cause the plate to differentiate easily (e.g., middle image in Fig. 10). This method is motivated from the nature of the license plate surface and has already been tested in many similar applications. License plates in Greece, as in many other countries in Europe and America, are made from a retroreflective material, which cause them to shine when the sun or even a bright light is directed toward them. This attribute makes license plates perfect candidates for cameras that are sensitive to infrared illumination. In those 44 images, the proposed algorithm identified correctly all the license plates and the characters in them. Motorbike license plates were not considered to our sample due to the fact that they are located in the rear part of the bike. More than one license plate can be segmented in the same image, as shown in Fig. 14.

Another aspect that imposes a restriction to a computer vision algorithm is the discrimination analysis that achieves. The minimum license plate in pixels that can be located by the LPR algorithm has been measured experimentally to 16×46 pixels. License plate frames that were below these dimensions were rejected by the license plate location part of the algorithm. On the other hand, for character segmentation and complete success in character recognition, an average plate resolution of 26×78 pixels was necessary. Below that limit, characters appear blurred and distorted when subjected to the

Fig. 14. Successful plate identification sequence in three nonmoving vehicles.

plate processing and character extraction part of the algorithm. Consequently, given that the license plate has a resolution of at least 26×78 pixels or more, successful identification of the plate and its characters is expected. As the resolution becomes lower, misclassifications would certainly occur.

VII. PERFORMANCE OF OTHER TECHNIQUES

There are several commercial LPR systems whose evaluation is beyond the review abilities of this paper due to the fact that their operation is strictly confidential, and moreover, their performance rates are often overestimated for promotional reasons. Table VI highlights important issues concerning various LPR systems presented in the literature. Issues such as processing time, computational power, and recognition rate are featured. In the last column, the scientific background for plate identification and OCR is indicated.

As already discussed, several other attempts have been made to implement LPR systems. Among the noncommercial systems that are presented in the literature, those portrayed in [4], [11], [12], [17], [18], [29], and [31] will be discussed in the following paragraphs. Those papers were selectively picked up for further comparison as the performance of the respective systems has been evaluated on more than 1000 test images of various illumination conditions.

Compared to the architecture presented in this paper, both the systems described in [29] and [31] report a better overall performance (94.5% and over 97%), respectively. However, in [29], as already discussed in Section II, the system sets strict limitations as far as the distance and angle of view is concerned, not to mention that a small portion of the vehicle images is examined.

Superior in terms of plate segmentation appears to be the approach presented in [4], where an impressive success ratio of 99.6% is achieved. Unfortunately, the latter work was restricted only to license plate segmentation. Among the most recent researches, in [18], a complete plate recognition ratio around 93% for 1065 views of 71 vehicles is reported. Those

TABLE VI
PERFORMANCE OF THE LPR SYSTEMS IN THE LITERATURE

Ref.	Recognition Rate pl: plate detection ch: char. recognition ov: overall success	Platform & Processor	Time (sec)	Scientific background P: License plate detection method R: Plate character recognition method
[2]	100% (pl) 92.5% (ch)	Matlab 6.0, P II, 300 MHz	~4	P: Connected component analysis. R: 2 PNNs (one for letter and one for numeral recognition). Test: 40 images.
[3]	80.4 % (ov)	AMD K6, 350 MHz	~ 5	P: Mathematical morphology. R: Hausdorff distance.
[4]	99.6% (pl)	P IV, 1.7 GHz	~0.1	P: Edge statistics and morphology. Test:~10000 images.
[5]	see scientific background	P IV, 2.4 GHz	~0.5	P: Horizontal - vertical edge statistics. 96.8% (first set), 99.6% (second set), 99.7% (third set).
[6]	94.2% (pl), 98.6% (ch)	C++, P II, 300 MHz	Not reported	P: Magnitude of the vertical gradients and geometrical features. Test: 104 images.
[7]	92.3% (pl), 95.7% (ch)	Not reported	~1.6	P: Block-based technique between a processing and a reference image. Motorcycle vehicles are also identified.
[8]	93.1% (ov)	Not reported	Not reported	P: Spatial gray-scale processing, R: dynamic projection warping.
[9]	90% (ov)	P III, 1 GHz	~0.3	P: Combination of color and shape information of plate.
[10]	97% (ov at day) 90% (ov at night)	Not reported	0.1 - 1.5	High Performance Computing Unit for LPR.
[11]	91% (ov)	RS/6000 mod. 560	~1.1	P: Contrast information. R: Template matching. Test: 3200 images.
[12]	80%(ov)	Intel 486/66MHz	~15	P: Grey scale processing. R: Feedforward ANN.
[13]	89% (pl)	C++, P III 800 MHz	~1.3	P: Support Vector Machine (SVM), continuous adaptive meanshift algorithm (CAMShift) in color measurements.
[14]	98.5% (pl)	AMD Athlon 1.2 GHz	~34	P: A supervised classifier trained on the texture features. R: "kd-tree" data and "approximate nearest neighbour". Test: 131 images.
[15]	95% (ov)	Visual C++, P IV	2	P: Multiple Interlacing method. R: ANN.
[16]	97% (pl)	Workstation SG-INDIGO 2	2	P: Fuzzy logic rules are applied for plate location only. Test: 100 images.
[17]	75.4% (pl), 98.5% (ch)	Not reported	Not reported	P: Fuzzy Logic. R: Neural network (Multi Layer Perceptron). Test: 10000 images.
[18]	97.6% (pl) 95.6% (ch)	P IV, 1.6 GHz	~2.5	P: Fuzzy Logic color rules. R: Self Organized Neural Network. Test: 1065 images of 71 vehicles.
[19]	98% (pl)	P IV, 1.4 GHz	3.12	P: Gabor Transform and Vector Quantization. Performs only plate location and character segmentation (94.2%).
[20]	Not reported	P III, 600 MHz	0.15	P: Sub-machine-code Genetic Programming.
[21]	80.6% (pl) average	P III, 500 MHz	0.18	P: Real-coded genetic algorithm (RGA).
[22]	Not reported	Not reported	Not reported	P: Markov Random Field and Genetic Algorithm in video sequence.
[23]	98.8% (pl), 95.2% (ch)	Visual C++, P IV 1.4 GHz	0.65 (plates) 0.1 (per character)	P: Hough transform. R: Hidden Markov Model.
[24]	95.6% (pl)	Not reported	Not reported	P: AdaBoost (Adaptive Boosting) meta - algorithm.
[25]	92.4% (pl)	Borland C++, P IV 1.6GHz	Not reported	P: Wavelet transform based method.
[26]	93.6% (pl)	Visual C++, AMD Athlon 900Mhz	~1.3	P: Generalized Symmetry Transform and Image Warping.
[27]	85% (pl)	Not reported	Not reported	P: Discrete Time Cellular Neural Networks. R: Multi-Layer Perceptron.
[28]	85% (pl)	Pulse-Coupled NN Processor	Not reported	P: Pulse Coupled Neural Network (PCNN).
[29]	100% (pl), 94.7% (ch)	Visual C++, P II	~1	P: 2 Time-Delay Neural networks (TDNNs). R: support vector machine (SVM)-based character recognizer. Test: 1000 video sequences.
[30]	98% (pl)	C++, P I, 200 Mhz	0.2	P: Vector Quantization (VQ) for picture compression and effective license plate location.
[31]	97.0-99.3%	Not reported	Not reported	P: Sensing system with a wide dynamic range. Test: 1000 images.
[32]	84.5% (ch segmentation)	Not reported	Not reported	Character segmentation: Feature vector extraction and mathematical morphology.
[33]	96.8% (ch)	Visual C++, P IV 2.4GHz	Not reported	P: Block based local adaptive binarization method.
[34]	88% (pl), 95.7% (ch)	Not reported	Not reported	P: Connected components. R: Hidden Markov Models.
[36]	99%	Not reported	Not reported	R: Character recognition-PNN. Test: 680 segmented license plate images.
[37]	Not reported	Not reported	~1.5	P: Gradient analysis, R: Template matching
[38]	98.2% (pl)	Not reported	Not	P: Contour tracking and ART1Neural Network. Test: 114 images.

Fig. 15. Distinguishing parts of ambiguous characters as proposed in [18].

vehicles were photographed in complex background, various environments (street, roadside, and parking lot), and different illumination, some of them with damaged license plates.

The remaining approaches reported similar [11] or lower recognition ratios [12], [17] compared to our system, when natural images were forwarded to them. Impressive character recognition results were stated in [17] when a multilayered feedforward neural network was used in a vast amount of test cases. On the other hand, the plate segmentation in the same system was not similarly efficient.

To summarize, the results and the characteristics of our implementation place it as a valid competitor among other noncommercial systems. Specifically, the proposed algorithmic structure was evaluated on a fixed location (sample set 1 and 2) as well as in images acquired in various views and illumination conditions (sample 3 and 4). In addition, it was tested on vehicle images from various European countries (sample 5). The recognition module is not country specific like many others, the plate segmentation module can handle more that one license plate at the same time, and the developed software has relatively low computational needs. The average processing time for both plate identification and character recognition is at most 0.3 s. Our system suffered from specific misclassification errors that are identified and discussed in detail in Section VI. Work still remains to be done in order to improve the neural network's recognition performance and overcome, as much as possible, common misclassification errors (I–1 and O–0) that still occurred when the segmented plates were introduced to a commercial OCR system.

Besides *a priori* knowledge that can be inserted in the recognition module as proposed in Section VI, rule-based methods could also be incorporated. A similar approach in distinguishing characters 8 and B was proposed in [18]. To overcome the misclassification, the authors predefined an ambiguity set containing the pairs of the misclassified characters. For each character in the set, the nonambiguous parts of the character were specified as depicted in Fig. 15. During character recognition, once an unknown character is classified as one of the characters in the ambiguity set, an additional minor comparison between the unknown character and the classified character will be performed. The comparison will then focus only on the nonambiguous parts of the character.

VIII. SUMMARY—FUTURE EXTENSIONS

The operation of an automated vehicle license plate recognition system was analyzed in this paper in terms of software and hardware aspects. Its operation is divided in two image-processing phases: the phase of license plate segmentation and the phase of license plate processing and character recognition.

The former has been addressed through the implementation of SCWs method for image segmentation, connected component analysis, and binary measurements. The latter task is addressed again through the implementation of the SCW method for image binarization in conjunction with a PNN for character recognition. Software and hardware issues were described, problems and restrictions of the proposed algorithmic sequence were discussed, and finally, experimental results were assessed and compared against LPR systems that were proposed in the literature.

LPR, as a means of vehicle identification, may be further exploited in various ways, such as vehicle model identification as well as undervehicle surveillance. For the vehicle model identification task, the license plate position could play an important role in segmenting a distinctive reference area of the vehicle's frontal view. We are currently undertaking research for extracting key points from the vehicle mask on the basis of license plate position and creating a digital signature for every vehicle model. Moreover, for undervehicle inspection, it is assumed that a template undervehicle image for each inspected vehicle has been archived into a database in advance. Based on the incoming vehicle license plate, the respective template image is retrieved from the database and then compared to the one acquired during real-time undervehicle inspection.

REFERENCES

[1] Last day of access (2005, Aug. 10). [Online]. Available: http://itsdeployment2.ed.ornl.gov/technology_overview/

[2] C. Anagnostopoulos, E. Kayafas, and V. Loumos. (2000). Digital image processing and neural networks for vehicle license plate identification. *J. Elect. Eng.* [Online]. *1(2)*, pp. 2–7. Available: http://www.medialab.ntua.gr/people/canag/journals.php

[3] F. Martín, M. García, and L. Alba, "New methods for automatic reading of VLP's (Vehicle License Plates)," in *Proc. IASTED Int. Conf. SPPRA*, Jun. 2002. [Online]. Available: http://www.gpi.tsc.uvigo.es/pub/papers/sppra02.pdf

[4] B. Hongliang and L. Changping, "A hybrid license plate extraction method based on edge statistics and morphology," in *Proc. ICPR*, 2004, pp. 831–834.

[5] D. Zheng, Y. Zhao, and J. Wang, "An efficient method of license plate location," *Pattern Recognit. Lett.*, vol. 26, no. 15, pp. 2431–2438, Nov. 2005.

[6] S. Z. Wang and H. M. Lee, "Detection and recognition of license plate characters with different appearances," in *Proc. Conf. Intell. Transp. Syst.*, 2003, vol. 2, pp. 979–984.

[7] H.-J. Lee, S.-Y. Chen, and S.-Z. Wang, "Extraction and recognition of license plates of motorcycles and vehicles on highways," in *Proc ICPR*, 2004, pp. 356–359.

[8] T.-H. Wang, F.-C. Ni, K.-T. Li, and Y.-P. Chen, "Robust license plate recognition based on dynamic projection warping," in *Proc. IEEE Int. Conf. Netw., Sensing and Control*, 2004, pp. 784–788.

[9] X. Shi, W. Zhao, and Y. Shen, "Automatic license plate recognition system based on color image processing," in *Lecture Notes on Computer Science*, vol. 3483, O. Gervasi *et al.*, Eds. New York: Springer-Verlag, 2005, pp. 1159–1168.

[10] D. Yan, M. Hongqing, L. Jilin, and L. Langang, "A high performance license plate recognition system based on the web technique," in *Proc. Conf. Intell. Transp. Syst.*, 2001, pp. 325–329.

[11] P. Comelli, P. Ferragina, M. N. Granieri, and F. Stabile, "Optical recognition of motor vehicle license plates," *IEEE Trans. Veh. Technol.*, vol. 44, no. 4, pp. 790–799, Nov. 1995.

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

[13] K. I. Kim, K. Jung, and J. H. Kim, "Color texture-based object detection: An application to license plate localization," in *Lecture Notes on Computer Science*, vol. 2388, S.-W. Lee and A. Verri, Eds. New York: Springer-Verlag, pp. 293–309.

[14] J. Cano and J. C. Perez-Cortes, "Vehicle license plate segmentation in natural images," in *Lecture Notes on Computer Science*, vol. 2652, F. J. Perales *et al.*, Eds. New York: Springer-Verlag, 2003, pp. 142–149.

[15] A. Broumandnia and M. Fathy, "Application of pattern recognition for Farsi license plate recognition," presented at the ICGST Int. Conf. Graphics, Vision and Image Processing (GVIP), Dec. 2005. [Online]. Available: http://www.icgst.com/gvip /v2/P1150439001.pdf

[16] N. Zimic, J. Ficzko, M. Mraz, and J. Virant, "The fuzzy logic approach to the car numlber plate locating problem," in *Proc. IIS*, 1997, pp. 227–230.

[17] J. A. G. Nijhuis, M. H. ter Brugge, K. A. Helmholt, J. P. W. Pluim, L. Spaanenburg, R. S. Venema, and M. A. Westenberg, "Car license plate recognition with neural networks and fuzzy logic," in *Proc. IEEE Int. Conf. Neural Netw.*, 1995, vol. 5, pp. 2232–2236.

[18] S.-L. Chang, L.-S. Chen, Y.-C. Chung, and S.-W. Chen, "Automatic license plate recognition," *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 1, pp. 42–53, Mar. 2004.

[19] F. Kahraman, B. Kurt, and M. Gökmen, "License plate character segmentation based on the gabor transform and vector quantization," in *Lecture Notes on Computer Science*, vol. 2869, A. Yazici and C. Sener, Eds. New York: Springer-Verlag, 2003, pp. 381–388.

[20] G. Adorni, S. Cagnoni, M. Gori, and M. Mordonini, "Access control system with neuro-fuzzy supervision," in *Proc. Conf. Intell. Transp. Syst.*, 2001, pp. 472–477.

[21] S. Yoshimori, Y. Mitsukura, M. Fukumi, and N. Akamatsu, "License plate detection using hereditary threshold determine method," in *Lecture Notes in Artificial Intelligence*, vol. 2773, V. Palade, R. J. Howlett, and L. C. Jain, Eds. New York: Springer-Verlag, 2003, pp. 585–593.

[22] Y. Cui and Q. Huang, "Extracting characters of license plates from video sequences," *Mach. Vis. Appl.*, vol. 10, no. 5/6, pp. 308–320, Apr. 1998.

[23] T. D. Duan, T. L. Hong Du, T. V. Phuoc, and N. V. Hoang, "Building an automatic vehicle license plate recognition system," in *Proc. Int. Conf. Comput. Sci. RIVF*, 2005, pp. 59–63.

[24] L. Dlagnekovin *License Plate Detection Using AdaBoost*. La Jolla: Comput. Sci. Eng. Dept., Univ. California San Diego, Mar. 2004. [Online]. Available: http://www.cse.ucsd.edu/classes/ fa04/cse252c/projects/louka.pdf

[25] C.-T. Hsieh, Y.-S. Juan, and K.-M. Hung, "Multiple license plate detection for complex background," in *Proc. Int. Conf. AINA*, 2005, vol. 2, pp. 389–392.

[26] D.-S. Kim and S.-I. Chien, "Automatic car license plate extraction using modified generalized symmetry transform and image warping," in *Proc. ISIE*, 2001, pp. 2022–2027.

[27] M. H. ter Brugge, J. H. Stevens, J. A. G. Nijhuis, and L. Spaanenburg, "License plate recognition using DTCNNs," in *Proc. IEEE Int. Workshop Cellular Neural Netw. and Appl.*, 1998, pp. 212–217.

[28] M. I. Chacon and A. Zimmerman, "License plate location based on a dynamic PCNN scheme," in *Proc. Int. Joint Conf. Neural Netw.*, 2003, vol. 2, pp. 1195–1200.

[29] K. K. Kim, K. I. Kim, J. B. Kim, and H. J. Kim, "Learning-based approach, for license plate recognition," in *Proc. IEEE Signal Process. Soc. Workshop, Neural Netw. Signal Process.*, 2000, vol. 2, pp. 614–623.

[30] R. Zunino and S. Rovetta, "Vector quantization for license plate location and image coding," *IEEE Trans. Industrial Electron.*, vol. 47, no. 1, pp. 159–167, Feb. 2000.

[31] T. Naito, T. Tsukada, K. Yamada, K. Kozuka, and S. Yamamoto, "Robust license plate recognition method for passing vehicles under outside environment," *IEEE Trans. Veh. Technol.*, vol. 49, no. 6, pp. 2309–2319, Nov. 2000.

[32] S. Nomura, K. Yamanaka, O. Katai, H. Kawakami, and T. Shiose, "A novel adaptive morphological approach for degraded character image segmentation," *Pattern Recognit.*, vol. 38, no. 11, pp. 1961–1975, Nov. 2005.

[33] B. R. Lee, K. Park, H. Kang, H. Kim, and C. Kim, "Adaptive local binarization method for recognition of vehicle license plates," in *Lecture Notes on Computer Science*, vol. 3322, R. Klette and J. Uniæ, Eds. New York: Springer-Verlag, 2004, pp. 646–655.

[34] D. Llorens, A. Marzal, V. Palazon, and J. M. Vilar, "Car license plates extraction and recognition based on connected components analysis and HMM decoding," in *Lecture Notes on Computer Science*, vol. 3522, J. S. Marques *et al.*, Eds. New York: Springer-Verlag, 2005, pp. 571–578.

[35] C. Anagnostopoulos, T. Alexandropoulos, S. Boutas, V. Loumos, and E. Kayafas, "A template-guided approach to vehicle surveillance and access control," in *Proc. IEEE Conf. Advanced Video and Signal Based Surveillance*, 2005, pp. 534–539.

[36] Y. Hu, F. Zhu, and X. Zhang, "A novel approach for license plate recognition using subspace projection and probabilistic neural network," in *Lecture Notes on Computer Science*, vol. 3497, J. Wang, X. Liao, and Z. Yi, Eds. New York: Springer-Verlag, 2005, pp. 216–221.

[37] Y.-P. Huang, S.-Y. Lai, and W.-P. Chuang, "A template-based model for license plate recognition," in *Proc. IEEE Int. Conf. Netw., Sensing and Control*, 2004, pp. 737–742.

[38] K.-B. Kim, S.-W. Jang, and C.-K. Kim, "Recognition of car license plate by using dynamical thresholding method and enhanced neural networks," in *Lecture Notes on Computer Science*, vol. 2756, N. Petkov and M. A. Westenberg, Eds. New York: Springer-Verlag, 2003, pp. 309–319.

[39] D. F. Sprecht, "Probabilistic neural networks," *Neural Netw.*, vol. 3, no. 1, pp. 109–118, 1990.

[40] C. M. Bishop, *Neural Networks for Pattern Recognition*. New York: Oxford Univ. Press, 1997.

[41] J. Sauvola and M. Pietikäinen, "Adaptive document image binarization," *Pattern Recognit.*, vol. 33, no. 2, pp. 225–236, Feb. 2000.

[42] S. E. Umbaugh, *Computer Vision and Image Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1998, pp. 133–138.

[43] E. Parzen, "On estimation of a probability density function and mode," *Ann. Math. Stat.*, vol. 33, no. 3, pp. 1065–1076, 1962.

[44] P. D. Wasserman, *Advanced Methods in Neural Computing*. New York: Van Nostrand Reinhold, 1993.

[45] I. Anagnostopoulos, C. Anagnostopoulos, V. Loumos, and E. Kayafas, "Classifying web pages employing a probabilistic neural network," *Proc. Inst. Electr. Eng.—Software*, vol. 151, no. 3, pp. 139–152, 2004.

[46] Last day of access, March 4th 2005. [Online]. Available: http://www. htsol.com/

[47] L. J. Nelson, "Video-based automatic vehicle identification," *Traffic Technol. Int.*, pp. 105–110, Jun./Jul. 1997

Christos Nikolaos E. Anagnostopoulos (M'04) was born in Athens, Greece, in 1975. He received the Mechanical Engineering Diploma from National Technical University of Athens (NTUA) and the Ph.D. degree from the Electrical and Computer Engineering Department, NTUA, in 1998 and 2002, respectively.

In 2003, he joined the University of the Aegean, Lesvos, Greece, as a Lecturer in the Cultural Technology and Communication Department. He has published more than 60 papers in journals and conferences. His research interests are image processing, computer vision, neural networks, and artificial intelligence.

Dr. Anagnostopoulos is a member of the Greek Chamber of Engineers.

Ioannis E. Anagnostopoulos (M'02) was born in Athens, Greece, in 1975. He received the Electrical and Computer Engineering Diploma from University of Patras, Patras, Greece, in 1998 and the Ph.D. degree from School of Electrical and Computer Engineering of the National Technical University of Athens (NTUA) in 2004.

In 2004, he joined the University of the Aegean Samos, Greece, as a Visiting Lecturer at the Department of Information and Communication Systems Engineering. He has published more than 60 papers in journals and conferences. His research interests are in the fields of web information management, E-commerce, communication networks, image processing, medical information systems, and multimedia systems and software.

Dr. Anagnostopoulos is a member of the Greek Chamber of Engineers.

Vassili Loumos (M'97) received the Diploma of Electrical Engineering and the Ph.D. degree in computer science from the National Technical University of Athens (NTUA), Athens, Greece.

He spent four years at the "Centre National de Recherhe Scientifique," Paris, France, working as a Research Engineer, and three years with CGEE ALSHTOM, working as a Field Engineer, before joining the academia. He became a faculty member of the School of Electrical and Computer Engineering of the NTUA in 1990. Currently, he is a Professor at the NTUA, teaching multimedia technologies and computer graphics. In parallel, he is running the Multimedia Technology Laboratory of the NTUA, which he founded in 1995. His research and development activities focus on the fields of image processing and computer vision, internet technologies and applications, and computer networks.

Prof. Loumos is a member of the Greek Chamber of Engineers.

Eleftherios Kayafas (M'97) received the B.Sc. degree from Athens University, Athens, Greece, in 1970 and the M.Sc. and Ph.D. degrees in electrical engineering from University of Salford, Salford, U.K., in 1975 and 1978, respectively.

He worked with the Civil Aviation Service in 1973–1974 and after his postgraduate studies with the Hellenic Aerospace Industry (HAI) in 1978–1979. In 1979, he joined the National Technical University of Athens (NTUA) as a Lecturer in the Electrical Engineering Department, then Assistant Professor in 1987, Associate Professor in 1992, and finally, Professor of applied electronics in 1996. He has published more than 140 papers in journals and conferences. His research interests are applied electronics, multimedia applications, and multimedia communication systems.

Prof. Kayafas is a member of the Greek Chamber of Engineers and the International Measurement Confederation and a member of the TC-4 Committee (IMEKO).