The CSI Journal on
Computer Science and Engineering
Vol. 7, No. 2 & 4 (b), 2009
Pages 51-67
Short Paper

Reliable License Plate Recognition

Aminollah Mahabadi Akbar Ranjbar

Department of Electrical Engineering, Shahed University, Tehran, Iran

Abstract

This paper presents a new method for the detection and recognition of the Persian-Arabic vehicle license plate system from outdoor images. The methods designed with multithreading architecture for implementation on a parallel architecture. The basis system uses symmetry for vehicle detection, line sweeping for plate detection and fuzzy method for plate recognition. The main theme is to use a novel Multilevel Illumination-Dim Model for image evaluation in such a way that the plate and its number can be extracted accurately with an inference Reliability Rate of Recognition. The method makes the extraction of the plate independent of color, size, and the location of license plate in a vehicle image. The camera didn't focus in the plate so the system is insensitive to camera angle of view and camera-vehicle relative position within a reasonable distance. The experimental results on 500 vehicle color images reached 98.68% for plate recognition, 100% for plate detection and 100% for in existence of vehicle or plate in the picture. Weather and lighting conditions, normal imaging distance, reasonable range of illumination, and angle of view didn't affect efficiency. Better performance with improvement in speed and accuracy has been reported while limitations in distance, angle of view, illumination conditions are set and background complexity is low.

Keywords: License Plate Recognition (VLPR), Reliability Rate of Recognition (3R), Persian Optical Character Recognition (POCR), Multilevel Illumination-Dim Model (MIDM), Intelligent Transportation Systems.

1. Introduction

License plate recognition (LPR) technique is the key to intelligent transportation systems (ITS) which consists of intelligent infrastructure systems and intelligent vehicle systems. The LPR consists of license plate detection (LPD), characters segmentation, and characters recognition. The most vital part of these system is the detection and extraction of the vehicle license plate (VLP) which directly affects the systems overall accuracy. The presence of noise blurring in the imago, uneven illumination, dim light and foggy conditions make the task even more difficult.

With market development, application increase, and cultural customization, the plate ideate feinting becomes more and more import ant. Since the needs (demands) for intellect systems are recently increased rapidly and the reliability is the most imported featnes of intelligent traffic systems, the Robust or Reliable Intelligent Vehicle License plate Recognition (RIVLPR) is main core of the research and developments. Low resolution and poor lighting caudation images of parking low cost cameras detection is a real challenge in automatic license plate recognition. The limited recorded character resolution accompanied with dirt, screws, bolts, overhanging vehicle parts and so forth, make the reliable optical character recognition (OCR) development very complicated specially the Persian type. As aren't, a lot of common approaches in conventions OCR systems [1] turn soul to be useless for RIVLPR. Therefore, developing a system which includes both rule defined by the license plate registration regulation [2] and OCR features [3] is essential especially in real time multinational VLPR [4-5].

The focus of this paper is on a novel Multilevel Illumination-Dim Model (MIDM) implemented in an RIVLPR system able to cope with outdoor as well as indoor conditions. Specifically, the novelties of contributions are:

1. A multilevel illumination-dim model used for extracting the plate and its number accurately under any lighting conditions and variant illumination.

2. A reliability model used for inference the reliability rate of plate detection and its number recognition.
3. An improved technique used for detecting vehicles in images.
4. A new technique able to improve character recognition during the VLPR operations.
5. A different adaptive thresholding method and dynamic programming approach were adopted for image binarization.
6. 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 with reasonable camera-vehicle distance.

In addition, a Persian Optical Character Recognition (POCR) system based on fuzzy technology is implemented, featuring the advantage of continuous improvement with high accuracy. The fuzzy architecture is based on a previous original project of our [1].

2. Related Work

The complexity of automatic vehicle license plate recognition varies throughout the world. In some countries, vehicle license plate design is highly standardized. Standard plate design makes it easier to detect and read vehicle license plates in images. It is therefore, not surprising that existing commercial applications have focused on countries with standardized VLP design. Where there are fewer strict specifications of VLP design, more sophisticated techniques are required for automatic VLP recognition.

This has made it difficult for existing commercial systems to be deployed in those countries. Persian vehicle license plate designs vary widely in all aspects providing a much more challenging environment for automatic VLP recognition, so we selected new Iranian plate structure. This plate type has many problems like different font size and stroke with low standard.

A reliable intelligent automatic VLP recognition solution typically addresses five key issues:

1. Vehicle presence: Is a vehicle present (only in our system used for LPD, and we call it vehicle detection)?
2. Plate location: Where is the VLP plate in the image?
3. Glyph location: Where are the VLP glyphs within the plate?
4. OCR: What are the characters on the plate (in our system, we call it POCR)?
5. Reliability: What is the reliability rate of recognition (an alarm message to the operator, only in our system presented, and we call it 3R and its message 3R-message)?

Recognition algorithms are generally composed of several processing steps, such as the detection of the vehicle, extraction of a license plate region, segmentation of characters from the plate and recognition of each character. Papers that follow these steps' frameworks are covered according to their major contribution in this section, without any research history on vehicle detection and Reliability Rate of Recognition (3R), in this era. The main goal of this section is to provide a brief reference in VLP identification and recognition, especially for Persian plates, regardless of particular application areas (i.e., stolen car search, traffic surveillance, etc.).

2.1. Vehicle Detection

While several systems have been developed to robustly detect instances of specific objects in natural images, the detection of generic object classes remains an unsolved problem in computer vision. Evidence from recent studies suggests that accurate and powerful object detectors can be built in a hierarchical manner, with lower level detectors first finding parts of the object, and a higher-level classifier combining their outputs.

In driver assistance applications, vehicle detection algorithms need to process the acquired images at real-time or close to real-time. Searching the whole image to locate potential vehicle locations is not realistic. The majority of methods reported in the literature follow two basic steps: 1. Hypothesis Generation (HG) where the locations of vehicles in an image are hypothesized and, 2. Hypothesis Verification (HV) where tests are performed to verify the presence of a vehicle in an image.

2.1.1. Hypothesis Generation (HG)

The objective of the HG step is to find candidate vehicle locations in an image quickly for further exploration. HG approaches can be classified into one of the following three categories: Knowledge-based, Stereo-vision based, and Motion-based.

Knowledge-Based Methods

Knowledge-based methods employ a priori knowledge to hypothesize vehicle locations in an image. We review below some approaches using information about *symmetry*, *color*, *shadow*, *corners*, *horizontal* or *vertical edges*, *texture*, and *vehicle lights*.

Symmetry: Vehicle images observed from rear or frontal view are in general symmetrical in horizontal and vertical directions. This observation was used as a cue for vehicle detection [6]. An important issue that arises when computing symmetry from intensity, however, is the presence of homogeneous areas. In these areas, symmetry estimation is sensitive to noise. In [7], information about edges was included in the symmetry estimation to filter out homogeneous areas. When searching for local symmetry, two issues must be considered carefully. First, we need a rough indication of where a vehicle is probably present. Second, even when using both intensity and edge maps, symmetry as a cue is still prone to false detections, such as symmetrical background objects, or partly occluded vehicles.

***Color*:** Although few existing systems use color information to its full extent for HG, it is a very useful cue for obstacle detection, lane/road following, etc. Several prototype systems investigated the use of color information as a cue to follow lanes/roads, or segment vehicles from a background [8]. Consequently the apparent color of a car varies under different ambient conditions.

Shadow: Using shadow information as a sign pattern for vehicle detection was initially discussed in [9]. Area underneath a vehicle is distinctly darker than any other area on an asphalt paved road [10], but the problem was the threshold variation of shadow intensity. In [11] a normal distribution was assumed for the intensity, but the

assumption about the distribution of road pixels might not always hold true, sine weather condition will change the road pixel shadow color, so making this method to fail.

Corners: Since the vehicles in general have a rectangular shape, the corner based method to hypothesize vehicle location is proposed [12]. Based on the four corners, four templates were used to detect all the corners in an image.

Vertical or horizontal edges: Different views of a vehicle, especially rear views, contain many horizontal and vertical structures, such as rear-window, bumper, etc. Using constellations of vertical and horizontal edges has shown to be a strong cue for hypothesizing vehicle presence. Authors in [13] applied a horizontal edge detector on the image first, and then the response in each column was summed to construct the profiles, and smoothed by using a triangular filter. By finding the local maximum and minimum peaks, they claimed that they could find the horizontal position of a vehicle on the road. A shadow method, similar to that in [11], was used to find the bottom of the vehicle. Authors in [14] proposed a method called Local Orientation Coding (LOC) to extract edge information. In [15] also used LOC, together with shadow information, for any vehicle detection.

Utilizing horizontal and vertical edges as cues can be very effective. However, an important issue to be addressed, especially in the case of on-line vehicle detection, is how the choice of various parameters affects system robustness. These parameters include the threshold values for the edge detectors; the threshold values for picking the most important vertical and horizontal edges, and the threshold values for choosing the best maxima (i.e., peaks) in the profile images. Although a set of parameter values might work perfectly well under some conditions, they might fail in other environments. The problem is even more severe for an on-road vehicle detection system since the dynamic range of the acquired images is much bigger than that of an indoor vision system. A multi-scale driven method was investigated in [17] to address this problem. Although it did not root out the parameter setting problem, it did alleviate it to some extend.

***Texture*:** The presence of vehicles in an image cause local intensity changes. Due to general similarities among all vehicles, the intensity changes follow a certain pattern, referred to as texture in [18]. Entropy was first used as a measure for texture detection. Another texture-based segmentation method suggested in [18] used co-occurrence matrices. Using texture for HG can introduce many false detection conditions. For example, when we drive a car outdoor, especially in some downtown streets, the background is very likely to contain textures.

Vehicle lights: Most of the cues discussed above are not helpful for night time vehicle detection. It would be difficult or impossible to detect shadows, horizontal or vertical edges, or corners in images obtained at night conditions. Vehicle lights represent a salient visual feature at night. Authors in [19] used morphological analysis for detecting vehicle light pairs in a narrow inspection area.

Stereo-Vision Based Methods

There are two types of methods using stereo information for vehicle detection. One uses a disparity map, while the other uses an anti-perspective transformation (i.e., Inverse Perspective Mapping).

Disparity map: The difference in the left and right images between corresponding pixels is called disparity. The author in [20] proposed a method of employing the power of the disparity while avoiding some heavy computations. Authors in [21], argued a local feature extractor “structure classification” was proposed to solve the correspondence problem easier.

Inverse perspective mapping: The term “Inverse Perspective Mapping (IPM)” does not correspond to an actual inversion of perspective mapping [22], which is mathematically impossible. Rather, it denotes an inversion under the additional constraint that inversely mapped points should lie on the horizontal plane. Assuming a flat road, authors in [23] used stereo vision to predict the image seen by the right camera, given the left image, using IPM.

Instead of warping the right image onto the left image, authors in [24] computed the inverse perspective map of both the right and left images. Authors in [25] investigated a trinocular system.

In general, stereo-vision based methods are accurate and robust only if the stereo parameters have been estimated accurately.

Motion-Based Methods

Three factors causing poor performance were summarized in [26]: (a) displacement between consecutive frames, (b) lack of textures, and (c) shocks and vibrations. Authors in [26] managed to remap the corresponding points between two consecutive frames, by minimizing a distance measure.

In contrast to dense optical flow, “sparse optical flow” utilizes image features, such as corners [27], local minimal and maximal [28], or “Color Blob” [29]. In general, motion-based methods can detect objects based on relative motion information. Obviously, this is a major limitation, for example, this method cannot be used to detect static obstacles, which can represent a big threat.

2.1.2. Hypothesis Verification (HV)

The input to the HV step is the set of hypothesized locations from the HG step. During HV, tests are performed to verify the correctness of a hypothesis. HV approaches can be classified into two main categories:

Template-Based Methods

Template-based methods use predefined patterns of the vehicle class and perform correlation between the image and the template. Authors in [16] proposed a hypothesis verification scheme based on license plate and rear windows detection using constraints based on vehicle geometry. Authors in [15] proposed a template based on the observation that the rear/frontal view of a vehicle has a ‘U’ shape.

Authors in [30] used a very loose template to recognize vehicles. In [31], the hypothesis was generated based on-road position and perspective constraints. The template contained a priori knowledge about vehicles: "a vehicle is generally symmetric, characterized by a rectangular bounding box which satisfies specific aspect ratio constraints."

Appearance-Based Methods

Appearance-based methods learn the characteristics of the vehicle class from a set of training images, which capture the variability in vehicle appearance. In [13], Principal Component Analysis (*PCA*) was used for feature extraction and Neural Networks (NNs) for classification. Authors in [14] used a method called Local Orientation Coding (LOC) to extract edge information. Authors in [18] designed two models for vehicle detection: one for sedans, and the other for trucks Moreover, the Hausdorrf distance was used for the classification of trucks and cars such as in [18].

A statistical model for vehicle detection was investigated by authors in [32]. They represented each vehicle image as a constellation of local features and used the Expectation-Maximization (EM) algorithm to learn the parameters of the probability distribution of the constellations.

In our work, we present a simple and speed car-detection method without any learning phase. The knowledge-based method extracts the central point between car (head or tail) lights based on central point between eyes extracted in [33]. Using this method, we have developed an algorithm that can reliably detect vehicles over a wide range of viewpoints.

2.2. License Plate Detection

As far as extraction of the plate region is concerned techniques based on combinations of edge statistics and mathematical morphology in [34-35] presented very good results. One disadvantage is that edge-based methods alone can hardly be applied to complex images, since they are too sensitive to unwanted edges.

Color or gray-scale based processing methods are proposed in the literature for license plate location [36-40]. The 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. In [41] a method is developed to scan a vehicle image with N row distance and count of the existent edges.

Fuzzy logic methods have been applied to the problem of locating vehicle license plates [42-44]. These methods are sensitive to the license plate color and brightness and need longer processing time from the conventional color-based methods.

Genetic Programming (GP) in [45] and Genetic Algorithms (GA) in [10-11] were also implemented for the task of license plate location. 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 [46] the authors acknowledged that the execution time of the HT requires too much computation, when applied to a binary image with a great number of pixels. In [47] a strong classifier was trained for license plate identification using the Adaptive Boosting (AdaBoost) algorithm.

A wavelet transform based method is used in [48] for the extraction of important contrast features used as guides to search for desired license plates. The major advantage of wavelet-transform is the fact that it can locate multiple plates with different orientations in one image. The method is unreliable when the distance between the vehicle and the acquisition camera is either too far or finally closes [49].

Symmetry is also used as a feature for car license plate extraction. The Generalized Symmetry Transform (GST) produces continuous features of symmetry. This process is usually time-consuming, and a rotated or perspective distorted license plate image is rather impossible to detect. In [50], the algorithm set limits to its effective distance and is insufficient when rotated or distorted plates appear [49].

In addition, various neural network architectures in [51-53] 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 our work, we present a rapid accurate method based on MIDM for plate detection task. We use symmetry in vehicle detection mainly for rapid plate locating combined with line sweeping and other techniques, counting the existent edges for plate detecting based on the plate's rules (e.g. abrupt changes in intensity in plate text). We make this method reliable by way of a few additional techniques. The method makes the extraction of the multiple plates independent of color, size, location of license plate in vehicle image, angle of view and the camera-vehicle distance. Weather and lightning conditions, variant illumination and angle of view didn't have any effect on efficiency.

2.3. Character Segmentation

Image segmentation may be defined as the partitioning of an image into regions or objects that have coherent characteristics (e.g., color and texture) which are meaningful for the application at hand [54]. Currently, there is no comprehensive theory of segmentation [55] and the usual approach consists of making use of a priori knowledge and ad-hoc procedures for each application. Known strategies include segmentation by fitting a model, segmentation by clustering, detection of boundaries, and region growing.

The license plate candidates determined in the previous stage are examined in the vehicle 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 [56] and Markov Random Fields [57].

In our work, we present a rapid segmentation method which utilizes a recursive flood fill algorithm like virus growing with a hit approach in [1], dynamic framing and normalization work. Before using a character, we need to normalize them. There are three reasons for normalization: 1. we can reduce the number of attributes, which will hasten the character recognition. 2. We will have the same number of attributes for all the characters because they will have the same dimensions. 3. We will improve the efficiency of the classifier as the input is more similar to the other instances in its class, and more different from the instances in other classes. The method can segment plate characters with various font size and stroke.

2.4. Thresholding

In many image processing applications, we often have to segment gray-level images into meaningful regions. In some cases, the image contains one or more objects and a

background. Image thresholding mainly serves as a tool for object recognition and feature extraction. There was no evidence that there is a best possible way to choose a value for threshold and therefore, this is to be decided according to the application [60]. Global thresholding is a general method for a region of interest (ROI) recognition. Adaptive local binarization methods are used [59, 61]. Unlike the global thresholding method, this approach computes an independent threshold for each pixel over a local window whose center is the pixel being binarized (*M*-window size).

In our work, we proposed two methods, global for plate extraction and local for plate recognition. The first is for improving the speed up of the whole system. The second is also based on an adaptive method for thresholding the image and the step of calculating local threshold value that reduces many redundant computations and improves the execution speed significantly. This method works only on a plate region, so it isn't slow. However different to the previous approaches, we do not use a criterion function to be minimized. Instead, the image is threshold based on knowledge about detection and recognition of the objects with MIDM parameters. Our method is the base on the statistical measurement and interaction with the MIDM for correcting the needed sets for ISMT.

2.5. Character Recognition

For recognition of segmented characters, numerous algorithms exploited mainly in optical character recognition applications utilized Hidden Markov Models (HMM) [46, 62], Neural Networks [41, 43-44, 63-65], Hausdorff distance [34], Support Vector Machine (SVM)-based character recognizer [53] and template matching [66, 39, 65].

The recognition results in [46, 62] were reported to be 92.5% and 95.7% after a complex procedure of preprocessing and parameterization for the HMMs [58]. In [43] a MLP was trained to recognize 36 characters of the English alphabet. The recognition, achieving excellent results (98.5%). In [44] 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, remarkable 95.6%. Probabilistic Neural Networks (PNN) for license plate recognition was introduced in [43]. In [65] was reported an impressive recognition rate that reached 99.5%.

Hausdorff distance is a method which compares two binary images. Its recognition rate is very similar to that obtained with neural network classifiers but it is slower. Additionally, in [56] a system implementing for Support Vector Machines (SVM) was designed and an average character recognition rate of 97.2% was reported. Properly built templates also obtained very good results for gray level images. A similar application is described in [39] for all the shifts of each character template over the sub-image containing the license plate.

In our work, we present an accurate technique based on fuzzy method to identify alphanumeric characters for Persian Optical Character Recognition (POCR) task. The method is also independent of thickness and size of font under different lighting conditions while it is robust to rotation for numerals and small rotations for alphabetic characters. In unrecognized character conditions, we use MIDM to correct the images on bad illumination thus improving the system performance.

2.6. Result Messages

After the characters segmentation and overall VLP Recognition, the license plate operator needs to know that the recognized VLP is reliable or not. This knowledge is very important for the operator. The operator reads the VLP messages and then can decide to accept or reject the system results. This process can be done automatically by the system without the need of any manual operations. It is also necessary and suitable for evaluation imaging. So, after this phase, if your image does not qualify but your system is able to get another one image then the operator can do it again. There is not any research history in other related works for a useful message to operator like our 3R-message.

The rest of this paper is organized as follows: Section 3 describes the proposed system that was used in this work while Section 4 discuses the reliability on recognition, Multilevel Illumination-Dim Model and thresholding method. Sections 5 and 6 describe vehicle detection and license plate detection. Our VLP extraction method is described in Section 7. Section 8 describes VLP recognition. In Section 9 we describe the experimental results and Section 10 concludes our work.

3. Proposed System

The proposed system for the involved input image compression and noise reduction using a wavelet transform such as preprocessing [67], detection of Vehicle in the image using circle frequency filter, extraction of VLP in wavelet bounds using edge information, character segmentation using flood fill algorithm with symbol normalization, characters recognition using fuzzy logic method and reliability rate of recognition using MIDM (see Figure 1).

In addition, it concentrates on input image evaluation, vehicle detection, reliability rate of VLP recognition, bad illumination or reflection reduction, and accurate POCR. Our method makes the extraction of the plate independent of color, size, location of license plate in vehicle image, angle of view and the reasonable camera-vehicle distance. Weather and lighting conditions, normal imaging distance, variant illumination and reasonable angle of view didn't have any effect on efficiency.

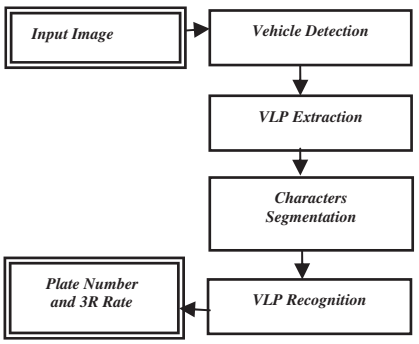

Figure 1. The proposed RIVLPR system

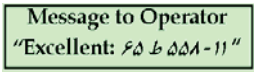

Figure 2. Reliable Rate of Recognition message to operator

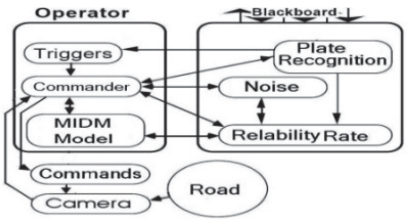

Figure 3. The RIVLPR system architecture

4. Reliability

E introduce a reliable multilevel illumination-dim model for improving the accuracy on plate detection, plate license recognition and the inferencing rate of reliability on VLP recognition. It depends on the detection and recognition levels of an object (e.g. VLP) in a subject (e.g. VLP detection and recognition). MIDM can extract the reliability rate on detection and recognition (3R) of an object. In addition, the model can be used for evaluating of input images, image thresholding, plate detecting, and characters recognizing such as [68].

The idea in using this model is for evaluating brightness or darkness levels of input image based on a knowledge feedback about detection and recognition states of VLP. The extracted knowledge has some parameters for modeling 3R based on the main VLP operations. The presented RIVLPR can operate in a wide range of illumination conditions with a benefit alarm message to operators about its work (see Figure 2) and commands to the camera for position correction or imaging condition. The relationship of the RIVLPR system components is presented in Figure 3.

4.1. Multilevel Illumination-Dim Model

We propose a simple scheme for quality evaluation of an input image and analyzing the system results. The gray scale illumination is considered as a feature for image evaluation. However, our method can be applied to any feature that should be separated. In order to remove foreground areas so small that they are unlikely to contain a vehicle, and also to remove noise and small outgrowths on larger areas, the foreground is first eroded (maximum three times in our work).

$$\alpha = \frac{\sum_{i=1}^{m_i}\sum_{j=1}^{n_i} pixel_{i,j}}{3 \times m_i \times n_i} \approx 0.6 (In \quad our \quad work) \tag{1}$$

$$g(x,y)_i = f(x,y)_i \times \alpha_i \therefore where \;\; \alpha_i = \alpha_{i-1}^{\;\;i-1} \tag{2}$$

The reason for using more than one step erosion is that it increases the chance for the license plate to be detected and recognized under bad lightning conditions. The architecture of the model includes: Gray Level Evaluator, Parameter Estimator using (1), Multi-Level Image Constructor using (2), Image Selector, Thresh older, Rules Evaluator, and Monitor. A subversion of the MIDM pseudo code is indicated in Table 1 and the respective flowchart is highlighted in Figure 4. The original image dimension is $m_i \times n_i$ of the i th image.

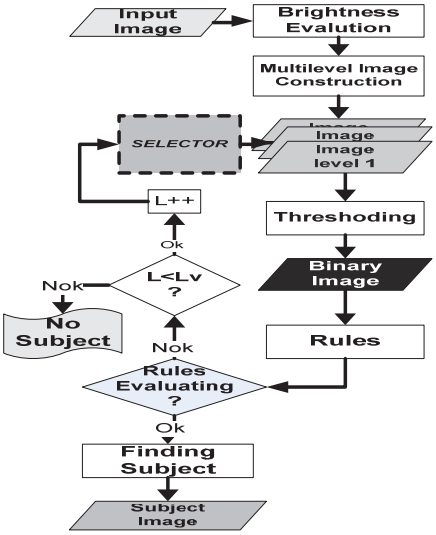

Figure 4. The MIDM architecture

4.2. Reliability of Recognition Model

The actual method for computing R_i as term 3R of the i_{th} observed image depends on the underlying model used, using (3), and taking the following factors into account: 1. $\alpha_i^{'}$, plate detection level in MIDM; 2. $\beta_i^{'}$, plate recognition level in MIDM; 3. $m_i \times n_i$, Original image height and width of the i_{th} observed image; 4. $m_i^{'} \times n_i^{'}$, Image plate height and width of the i_{th} observed image; 5. $K_i^{'}$, Image compression ratio (in our work is equal to %75) and it is not equal to %100; 6. $K_i = \sqrt{m_i \times n_i / m_i^{'} \times n_i^{'}}$, the aspect ratio of original input image to plate image (original input image/ plate image ratio. Where $m_i \times n_i$ the dimension of original is input image and $m_i^{'} \times n_i^{'}$ is the dimension of plate image). The maximum and minimum rate of 3R is with $\alpha_i^{'} = 1, \beta_i^{'} = 1$ (best image) and $\alpha_i^{'} = 3, \beta_i^{'} = 3$ (worst image) of the i_{th} observed image, respectively.

$$R_i = 100 - 0.577 \times \left(\frac{2^{\,k_i - 1 / k_i}}{1 - K_i^{'}} \times (\alpha_i^{'2} + \alpha_i^{'} \beta_i^{'2} + \beta^{'3}) \right) \tag{3}$$

4.3. Thresholding

A method is presented for adaptive vehicle image binarization, where the page is considered as a collection of subcomponents such as VLP text, background and picture. The problems caused by noise, illumination and many source type related degradations are addressed. A non-uniform illumination destroys the reflectance patterns that can be exploited by thresholding (e.g. plate extraction). The reflectance depends on car lights brightness and weather condition (or day light). A new method is applied to determine a local threshold for each pixel.

The performance evaluation of the algorithm utilizes real images with truth and inference, evaluation metrics for binarization of real images, and a weight-based ranking procedure. The final presented result includes a message to the operator. The proposed method is tested with images including different size and illumination plate. The results show that the method adapts and performs well in all cases.

Hypothetically, vehicle 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 by the presence of a vehicle license plate. Based on the above, this paper proposes a new adaptive thresholding technique named Illumination Set Measurement Thresholding (ISMT).

Table 1. The subversion of MIDM steps

1 **For** j=1:*Lv*, where *Lv* is the number of erosions, in our work *Lv*=3
{ Input: image *I(j)*, parameter: m_i, n_i are height & width of image;

2 **For** each pixel in image *I(j)*,
{ **Estimate** erosion number parameter using the image *I(j)* with parameters: m_i, n_i ; }
Name the resulting parameter α_j ;
Calculate parameters: α_l set , τ_t set;
3 **For** each pixel in image *I(j)*, { **Multiply** each pixel at α_j ; }
Calculate the $\alpha_{j+1} = \alpha_j^{\,j}$; image erosion parameter
4 **Select** scenario or appropriate parameters;
5 **For** each pixel in image *I(j)*,
{ **Binaries** *I(j)* using Multi level thresholding model with parameters: α_l set , τ_t set; }
Name the resulting image *Ib(j)*;
6 **Employ** Fuzzy set rules on image *Ib(j)*;
7 **If** (result is OK) {**Find** interest parameters; **Print** results; **Name** the resulting subject image *IS(j)*; **Exit**;}
Else { **If** (j > n+1) { **Print** "No subject found"; **Exit**; }
}

The method is developed in order to describe the "local" irregularity in the image using image statistics value. The algorithm was developed implementing the following steps:

1. Creation of two concentric windows $A_{m\times n}$ and $B_{2m\times 2n}$ pixels. The windows are presented in Figure 5a.
2. Calculation of statistical measurements in windows A & B.
3. Selection of two sets (threshold and brightness) from statistical measurements based on *ISMT* technique.
4. Selection of optimum threshold value based on knowledge about plate detection and recognition.

Definition of a rule: If the levels of plate detection and recognition in MIDM are stored in α and β respectively, then we select different values set from windows A and B. The two windows slide until the entire image is scanned as shown in Figure 5b. Then the central pixel of the windows is considered to belong to a plate. Based on α and β values and brightness set, we set threshold value from the statistical measurements in the windows and two sets $(\tau_w \,\&\, \tau'_w)$.

So, there are two scenarios for selecting a threshold value, *well* for good image and *worse* for bad image according to (4). The threshold value (τ_i) is selected based on $\{\alpha_{l0},...,\alpha_{lm}\}$ and $\{\alpha_{f0},...,\alpha_{fm}\}$ brightness sets, by using (5) based on Table 2 and (6) based on Table 3. Where, α_i and $p_v[j]$ are the gray level of each pixel and a counter for counting the number of each pixel with a gray level, respectively.

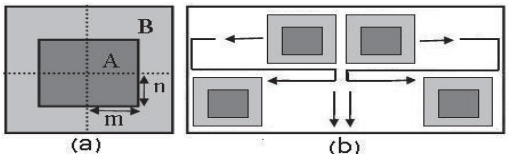

Figure 5. The concentric windows A and B shapes with (a) location and (b) movement

Table 2. Threshold value setting in the worst case scenario

1 **For** k=1:*Fn*, where *Fn* is the number of frames
{ Input: image frame *I(k)*, parameters: m_f, nf , are height & width of the frame; }

2 **For** each pixel in the image frame I(k),
{ **If** ($\alpha_{l0} < pixel\ (i, j) < \alpha_{l2}$) $P_v[0]++$;
If ($pixel\ (i, j) > \alpha_{l1}$) $P_v[1]++$;}
3 **If** ($P_v[1] \geq P_v[0]$) {**Set** $\tau_{(i,j)}$ to τ_0 ; **Exit**;}
Else {**Set** $\tau_{(i,j)}$ to τ_1 ; **Exit**; }

$$\tau_i \subseteq \begin{cases} \tau_w = \{\tau_0,...,\tau_{nl}\} & \therefore\ well :: if(\alpha \wedge \beta) = 1 \\ \tau'_w = \{\tau_0,...,\tau_{ml}\} & \therefore\ worse :: if(\alpha \vee \beta) > 1 \end{cases} \quad (4)$$

$$\tau_i \subseteq \tau w = \{\tau_0, \tau_1\} \ ::if \quad \alpha_s < p_v[j] < \alpha_z \ Where \ \ \alpha_i \in \{\alpha_{l0}, \alpha_{l1}, \alpha_{l2}\} \quad (5)$$

$$\tau_i \subseteq \tau'_w = \{\tau_0,...,\tau 4\} \ ::if \quad \alpha_s < p_v[j] < \alpha_z \ Where \ \ \alpha_i \in \{\alpha_{f0}, \alpha_{f1}, \alpha_{f2}\} \quad (6)$$

The parameters *ml, nl, fm* and *lm* could be updated according to the specific application. However, after experimentation in the original ISMT method, it was interestingly found that if the center of the concentric windows is near the center of an object to be segmented, then the results will be optimal. Therefore, the parameters $ml = 5$, $nl = 2$, $fm = 3$ and $lm = 3$ were selected on the basis of the above observation. In our work, we select and set the threshold value based on the gray level domains like Figure 6. Where *L* is the level of an operation in MIDM model (*L* is equal to α or β).

As the vehicle license plate in IRAN have an aspect ratio (height/width) less than 1/5 (or (width/height)>5), the Xs in the concentric windows were selected to be 2 times bigger than the Ys. The value of the threshold depends upon range fitting parameters and 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 and y. So, let x, y be the coordinates of the examined pixel in inspected image. This method adapts the threshold according to the local range over a window size of $n\times m$. The threshold at pixel (x, y) is calculated as (7).

Table 3. Threshold value setting in well case scenario

1 **For** k=1:*Fn*, where *Fn* is the number of frames
{ Input: image frame *I(k)*, parameter: m_i, n_i , are height & width of the frame;}

2 **For** each pixel in the image frame *I(k)*, where $P_v[i]$ is a counter
{ **If** ($\alpha_{l0} < pixel\ (i, j) < \alpha_{l1}$) $P_v[0]++$;
If ($\alpha_{l1} < pixel\ (i, j) < \alpha_{l2}$) $P_v[1]++$;
If ($pixel\ (i, j) > \alpha_{l2}$) $P_v[2]++$;
}

3 **For** i=0:*Cn*, where *Cn* is the number of counters and $\tau_{(i,j)}$ is the intermediate threshold value , $\tau_{(i,j)}$ set to zero
If ($\tau_{(i,j)} \le P_v[i]$) {**Set** $\tau_{(i,j)}$ to $P_v[i]$;}

4 **For** i=0:*Cn*, where *Cn* is the number of counters
If ($\tau_{(i,j)} = P_v[i]$) {**Set** $\tau_{(i,j)}$ to τ_i ; **Exit**;}

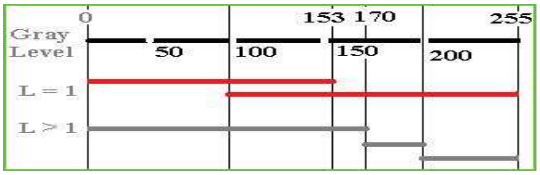

Figure 6. The gray level regions for decision making

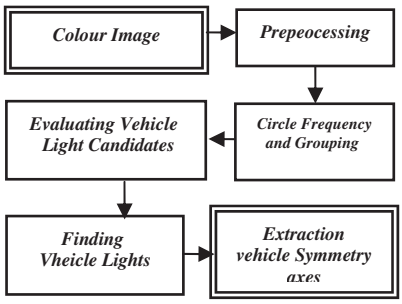

Figure 7. The RIVLPR Vehicle detection algorithm

$$T(x,y) = \left| \left(m(x,y) + \frac{1}{2}\left(\frac{\sigma(x,y)}{R} + 1 \right) - \tau_{(x,y)} \right) \right| \tag{7}$$

where in (7), $m(i, j)$, $\sigma(i, j)$ and R are the local sample mean, variance and rounding parameter (in our work $R = 3$), respectively. Experimental results suggest the values of $\tau\{\}$ which were adopted in this algorithm. For example in the case of badly illuminated areas, the threshold set is lowered. Hence, the pixel value in the respective coordinates x, *y* of the resulting image I_T is set either 0 (there is no plate) or 1 (there is a plate) according to (8).

$$I_T(x,y) = \begin{cases} 1 & if(I(x,y) \ge T(x,y)) \\ 0 & if(I(x,y) < T(x,y)) \end{cases} \tag{8}$$

5. Vehicle Detection

Vehicle detection is the most important and vulnerable stage of the whole process. Its goal is to deliver an accurate vehicle clip. To achieve this, the vehicle's image undergoes a series of processing steps such as preprocessing, circle frequency, grouping lights candidate, finding vehicle lights and extraction vehicle symmetry axes (see Figure 7).

5.1. Preprocessing

The preprocessing has three processing steps such as Gray scale image converting, wavelet transformation, and thresholding the image after selecting LL bound of wavelet result (see Figure 8). Preprocessing is designed to facilitate the further image analysis. The original image might be quiet large and will require a lot of processing time.

To avoid this, after getting the input color image and converting it to a gray scale image, we down-sample the input image to a quarter size of its original image by wavelet transform (see Figure 9). That is a good tool for data approximation, compression, noise removal and edge detection [24-25].

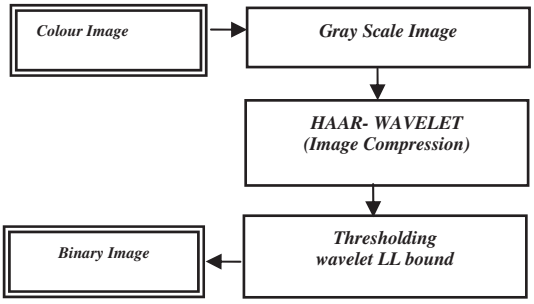

Figure 8. The Preprocessing step of RIVLPR

Figure 9. Hear-wavelet transforms image and channels

We would like to work on a down-sampled version of the image because of the scope which becomes smaller and many contiguous regions become physically connected. However, the resolution level we choose to work on has to meet the following constraint: any two heterogeneous regions in the original image should not be merged in the 1th resolution level [70].

We will apply here the discrete Hear-wavelet transform, representing the original pattern on the limited number of levels, representative for the task to provide the appropriate invariance's to the translation, scale and rotation [35] and enable good representation of the local features of the patterns. In the method, the result two-dimensional image has four different channels which are called low-passed, horizontal, vertical and diagonal, respectively. The transformed values in each channel occupy a quarter size of the original image.

Placed together, they look like what is depicted in Figure 9. The LL channel is a smoothed and down-sampled version of the original image. As a consequence, vertical edges or strokes would react strongly (namely, their transformed values are very positive) in the HL channel, and horizontal

edges or strokes in the LH channel. Diagonal strokes, on the other hand, would react in both H and V channels but their reactions are less vigorous than horizontal or vertical edges or strokes in HH channel. The next step is thresholding that transform the image to a binary conversion image. This method is explained in Section 4.3.

(a) (b)

Figure 10. (a) The original image and (b) CF-Filtered image

5.2. Vehicle Symmetry

Vehicle recognition plays an important role in the advancement of LPR since it provides a natural and efficient interface to camera. It is the best way for image capturing analysis and starting its process steps. For VLPR, vehicle detection is the first step. We can take different strategies for vehicle detection and plate recognition. This is based on the finding that once a vehicle is detected next step appears easier.

Our goal is to construct a fast vehicle detector by automatically extracting the symmetry feature and a vehicle validate. We chose to center our feature on the key point detections returned by the algorithm described in [33], because it is robust, computationally efficient, and independent of vehicle-camera distance and invariant to a number of common outdoor images transforms. We take the following approach. First we extract plate candidates from vehicle images. The next step is grouping candidates and selects the top five maximum points. Then the evaluating candidates step finds these two symmetry points. In the final step we find between lights.

In a typical vehicle region, many features are considered for vehicle detection such as the windshield and/or their corners, the logo, and the plate and/or its corners [71-72]. Among them, the shapes of the windshields, logo and plate can change drastically according to car types. The windshields, in contrast, are very stable, but the observable vehicle orientations relative to the camera direction are very limited. We therefore believe that these features are not suitable for vehicle recognition.

We try to start with a part other than the windshields, the logos or the plate. What is common to most vehicles and easy to find for a wide range of vehicle orientations? We claim that one possible candidate is the point between the lights (see Figure 10a). We call this point "Between-the-Lights" or "symmetry axes". The Between-the-Lights has two parts on both sides including the vehicle lights and it is comparably different from other sections. This is not different between the vehicle head lights or the tail lights. It can be seen for a wide range of vehicle orientations.

Moreover, the pattern around this point is fairly stable for any vehicle type. We use an image filter, i.e., the circle frequency filter (CF-filter), to detect Between-the-Lights (see Figure 10b). The CF-filter not only robustly extracts Between-the-Lights, but also many other similar characteristic points. Assume that $f_k = (k = 0,\ldots,N-1)$ are pixel values along a circle of radius r centered at (x, y). Then, the CF-filter calculates intensity $f(x,y)$ and phase angle $\phi(x,y)$ using (9) and (10). Here, f_k goes from the top $(x, y-r)$ along the circle in the counter-clockwise direction. Eq. (9) gives the spectrum power of the frequency with two terms calculated by the discrete Fourier transform of data series f_k, and Eq. (10) gives its phase angle.

$$f(x,y) = \left(\sum_{k=0}^{N-1} f_k \sin(4\pi k/N)\right)^2 + \left(\sum_{k=0}^{N-1} f_k \cos(4\pi k/N)\right)^2 \tag{9}$$

The Between-the-Lights has dark parts on both sides including lights and it is comparably bright on the upper side (forehead). If we draw a circle of a certain radius centered at Between-the-Lights, we can see two cycles of image intensities along this circle as bright–dark–bright–dark. When the center of the circle goes away from Between-the-Lights, the two-cycle pattern gradually collapses.

$$\phi(x,y) = \tan^{-1}\left(\frac{\sum_{k=0}^{N-1} f_k \sin(4\pi k/N)}{\sum_{k=0}^{N-1} f_k \cos(4\pi k/N)}\right) \tag{10}$$

We can therefore expect the intensity output of the CF-filter to become the local maximum at Between-the-Lights. When an image rotates in the image plane, the intensity output of the CF-filter does not change. This is a well-known Fourier transform characteristic. On the other hand, its phase angle output shifts exactly the same angle as the image rotation. Figure 11 shows an example of f_k s at Between-the-Lights (the logo at the center).

The radius of the circle is 7 pixels, and the 40 pixel values along it are plotted on the graph. The plot starts at the forehead and goes counterclockwise. There are two cycles of dark and bright. Figure 12a and Figure 12b show a CF-filtered image (in intensity) of Figure 11 and seven candidates of Between-the-Lights, respectively. At the peripheral, we cannot calculate Eq. (9), so we put the output as 0, where the width corresponds to the radius of the filter.

The top five local maximum points in the CF-filtered image are extracted as candidates of Between-the-Lights (maximum five points or less). Their positions are shown on the CF image. This means that we have to select the true Between-the-Lights from among several candidates. We try to find light-like regions on both sides of each candidate to confirm Between-the-Lights. However the conditions for these light-like regions are too simple for real-time execution.

In this paper, we propose a different grouping technique for selecting the true Between-the-Lights from among several candidates. In general, a grouping technique is weak in object rotations. With the CF-filter, however we can get not only an intensity output, but also a phase angle output. From this angle information, we can de-rotate the input image around each candidate point. This enables us to use a grouping technique for candidate evaluation.

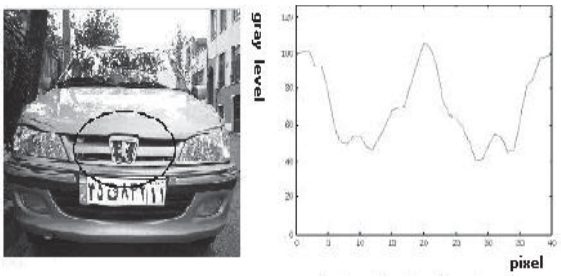

Figure 11. A plot of f_k

Our algorithm can detect Between-the-Lights correctly for all vehicles in different vehicle-camera distances (see Figure 13). At last, we search a symmetry pattern for finding lights. Vertical axes could be drawn between these lights making these axes a good place for finding a plate rapidly. It is not important that found symmetry belongs to lights or other features. If there is any plate around these axes, improvement of speed up of 30% in all RIVLPR Steps can be seen. After finding, we assume there is a car in the image.

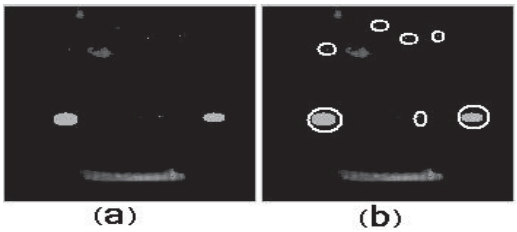

Figure 12. (a) CF-filtered of Figure 11 and (b) highlighted areas.

Figure 13. Different Car samples and its CF-Filter.

6. License Plate Detection

The first processing stage in a vehicle license plate recognition system is the detection and extraction of the vehicle license plate area from a larger scene image in order to minimize subsequent computations and algorithm complexity. Several approaches may be taken towards solving this problem, each having different computational expense and success rates. It is the purpose of this section to describe a set of applying techniques and evaluate their performance within the RIVLPR framework.

We use symmetry in vehicle detection mainly for rapid plate locating combination. Then by line sweeping and other techniques, counting the existent edges for plate detecting based on plate's rules (e.g. abrupt changes in intensity in plate text). We make this method reliable by a few additional techniques.

First, all plate candidates are evaluated purely (see Figure 14). The candidates can be a plate or similar. These are evaluated by line sweeping based on abrupt changes in vehicle image. Any weak candidate must be deleted. If the height of each area is less than 13 pixels, the area is not a plate and should be ignore. In the plates, each character has about 12.5% of plate length in average. If there is not any change in 1/3 of each area, the area will be deleted (see three steps in Figure 15).

Second, all strong candidates must be evaluated by plate rules. In a plate, the number of abrupt (or hit count) is about 28 transitions from black to white and vice versa (see Figure 16). In about 80% of conditions, the strong candidates found around the symmetry axes. Thresholding is done based on Section 4.3 method explained in Table 4 and 3R message produced, as all described steps under MIDM is performed.

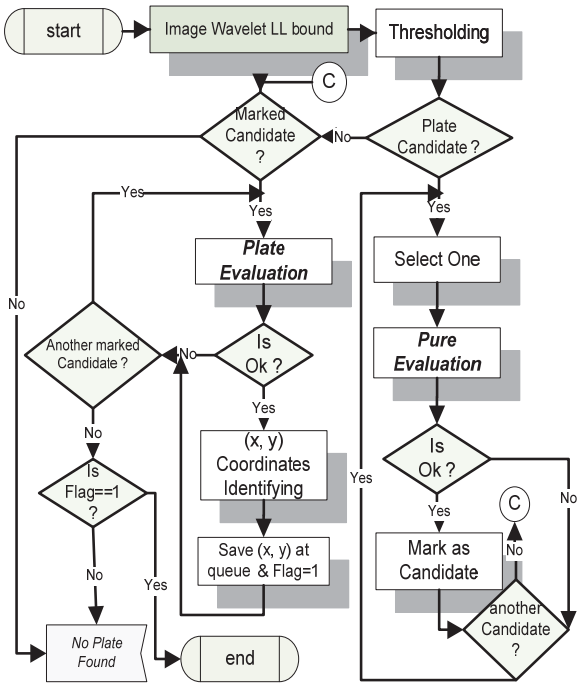

Figure 14. The license plate detection algorithm

Figure 15. Plate detection steps example

Figure 16. Abrupt changes in plate Persian text (K hits)

Table 4. The Vehicle license plate Detection steps

1 **Input** image IS
{**Parameter**: $m_i^{'}, n_i^{'}$ = plate image height, width;
(x, y) coordinates of first pixel.
Name the image IS as IP;
2 **For** all pixel in image IP,{ **Calculate** parameters: α_l set , τ_t set;}
3 **For** each pixel in image IP,{**Binaries** IP with parameters: $m_i^{'}, n_i^{'}$; }
Name the resulting image IT;
}

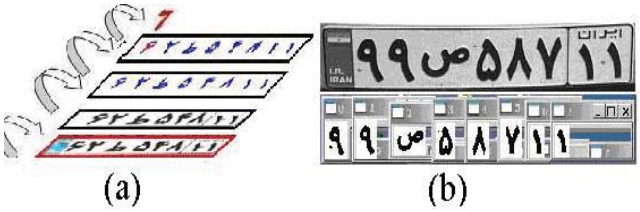

Figure 17. (a) Character extraction scheme and (b) a sample

7. Character Segmentation

The complicated step in VLPR before plate number recognizing is rapid character extracting from background in various environments light. Some text reading systems use segmentation schemes prior to OCR, but these schemes assume specific document layouts and require clean binary input. In systems where grayscale input is admitted, local thresholding is applied prior to recognition, which usually fails when backgrounds are small and not clean. So, the segmentation stage input is the original plate image so we first do the local plate image thresholding step based on what we explained before (see Section 4.3).

The method for character segmentation is done in four steps such as detecting, isolating, dynamic framing and normalizing respectively. So, they extract several parameters for segmentation and normalization steps. For many reasons the plate characters may have some interferences (e.g. dirt that connects two adjacent characters), therefore we cannot extract a character from the image with any horizontal or vertical sweeper. Here we use a recursive flood-fill algorithm to extract (detect and isolate) the character candidates from plate image and then framing it. The sweeper moves inside the image and as it hits the first black point of a character, it starts growing the black point to cover all the characters shape recursively and next will frame it. The black point must be adjacent to a white point for starting to find font stroke.

After framing the character, the frame analyzes for finding font size and stroke. If there is any unreasonable stroke in the frame, the area must be analyzed for character connectivity to a preserve adjacent one. The sweeper can find and memories the number of pixels as a stroke by frequent right movements. The goal of the presented algorithm is to find areas of similarly colored pixels with the same size and stroke, surrounded by pixels of contrasting color. For instance, a black character surrounded by a white border.

For each pixel in the plate image, the level of color is compared to the corresponding color level of the four pixels to the right, to the left, above and below of the examined pixel (four-way connectivity), recursively. If two adjacent pixels are the same color with reasonable stroke, they are considered to be connected and thus are given the different color as visited pixel [73]. Figure 17 shows the extraction scheme (see Figure 17a) and a sample segmented plate (see Figure 17b).

For each symbol, two sets of information, the number of hits, has been stored, one for vertical and the other for horizontal segment based on [1]. One of the most important advantages of hit counting is elimination of pen thickness, which reduces preprocessing efforts. The size of symbols (strokes) has always created some problems in optical character recognition systems. To overcome this problem we normalize the obtained features. After framing and normalizing, the area must be analyzed for a candidate validating, segmenting and extracting as a character [1].

After labeling the black and white spots the process of elimination takes place. The aim of this stage is to leave in the picture only these spots which are most likely to be vehicle license plate characters, assuming that each spot represents a single character and isn't connected with any other symbol in the image plate. Values of properly chosen parameters, which are evaluated and compared with certain expected empirically calculated values, are calculated. The spot is eliminated if one of the conditions given below is fulfilled (where $\mu_i = 1/k_i$ is the aspect ratio of plate image to original input image):

1. The width of the spot is smaller than $45/\mu_i$ points,
2. the width of the spot is bigger than 1/5 of the input image width,
3. the height of the spot is smaller than $90/\mu_i$ points,
4. the height of the spot is bigger than 0.8 of the input image height,
5. the ratio of the spot's width to its height is smaller than 0.1,
6. the ratio of the spot's width to its height is bigger than 4 (because of two or three joined characters), and
7. The ratio of the spots area to the spots' bounding box area is smaller than 0.15.

8. Message to Operator

After generating the RIVLPR message (i.e. 3R-message) the operator can read the message and decide to accept the results. So, in our method, it is necessary to define the reliability rate on recognition (3R), to evaluate 3R, to calculate it, to map on operator needs and to produce convenient messages to the operator.

For generating or producing this knowledge, we have done the process in four steps as 1- parameter extraction on plate detection and recognition (Section 4.2), 2- 3R calculation (Section 4.2), 3- mapping the 3R domain on the fuzzy message domain using (12), and 4- generating the message for operator knowledge. Message frame structure is as "Fuzzy decision or recognition rate or degree: Vehicle Plate Number" i.e. "Excellent: ۴۴ ق ۱۴۴-۱۱" (see Figure. 18).

Rilability Rate of Recognition -3RMSG	Vehicle Plate Number

Figure 18. System message frame structure, 3MSG: VPN

Figure 19. New Iranian Vehicle license plate

Fuzzy message domain: Based on needed reliability in a system, the presented message to the operator can be expanded or divided to many types but it is not necessary in our system. We divided this domain into four division domain *Excellent, well, Bad and Worst* based on human operator understanding about 3R (as (11)) by using (12). If the first section of the message frame structure contains "excellent" or "worst", the result of RIVLPR will be accepted or reject totally (see Figure 2).

$$\textit{Message} \quad Set = \{\text{"}Excellent\text{"}, \text{"}Well\text{"}, \text{"}Bad\text{"}, \text{"}Worst\text{"}\} \tag{11}$$

$$3RMSG = \begin{cases} \text{"}Excellent\text{"} : If\,(3R \geq 90) \\ \text{"}Well\text{"} : If\,(70 \leq 3R < 90) \\ \text{"}Bad\text{"} : If\,(50 \leq 3R < 70) \\ \text{"}Worst\text{"} : If\,(3R < 50) \end{cases} \tag{12}$$

9. Experimental Results

There are some international, national or local standards for VLP [74-75]. In Iran, the new VLP defines the country, the place of registration, etc. In this paper, the new Iranian plates are represented. The plate typically begins from the left with two numerals (*NN*), one letter (*L*) then three numerals (*NNN*) followed by another two numerals (*nn*) with different size and stroke. There is only one letter in a plate (one of eight characters), so accurate digit recognition is very important. Thus, the configuration of a common Iranian vehicle license plate is digit based in the form of "*NNLNNnn*".

For the purposes of vehicle licensing each province in Iran has an identifying character, vehicles registered in a particular province will therefore bear a vehicle license plate beginning with that province's identifying characters *"I.R.IRAN"* and national flag image in a blue box (see Figure 19). Additionally, certain divisions of government such as Military have their own distinguishing background color.

The experiments were conducted on 500 outdoor color images taken under different conditions including strong light, weak light, fade plate, uneven illumination and noisy plate. There is one natural image for different kind of vehicle (truck, car, van). The images were taken street lots without the vehicle initially coming directly towards the camera and didn't need to maximize the VLP size pixels. The images in the database have the following general characteristics: Each of the images is 1024x768 JPEG; in some of them, there are vehicles without any plate and hence there is no VLP to recognize.

Notice that the lengths of the plates are also different and the size (height and width) of the characters in the plate are not the same. The plate's color can be any color except the same as background. The vehicle-camera distance varied from 0.4 up to 6 meters at a height of 0.6-1.8 meters from the ground. The camera didn't focus on the plate so the system was insensitive to the camera's angle of view. Weather conditions and lightning, and illumination are different (see Figure 20). The VLP pictures used for this study were taken from the rear and front view of the vehicle. For the purpose of calculating the accuracy of this method, the total number of vehicle license plates in the database was counted and also the value used as this method's target. Note that the results are not for individual character recognition but for all the characters in the VLP. Therefore, to classify a vehicle license plate correctly all the characters and numerals in it must be correctly recognized.

Figure 20. Difficult car samples from database images

Figure 21. The RIVLPR user interface

The computation time of the proposed method is about 0.8 seconds with the worst illumination (MIDM level 3) and 0.6 seconds with the best illumination (MIDM level 1), if it is running on a Pentium-4 2.0GHz, 512MB RAM PC. The images were acquired using a low end Samsung gmax301 digital camera. The image processing algorithmic steps were completely built in Microsoft's Visual C++6.0 as a stand–alone executable program on windows XP. Figure 21 demonstrates a screen capture of the vehicle license plate recognition Graphical User Interface (GUI) in use.

As far as overall performance calculation is concerned, this can be achieved by calculating the percentage of vehicle license plates that have been correctly identified by the machine and verified by a person supervising the test sample. In addition, all results of the tables are the average execution for more than 70 runs of the algorithm (using an internal microsecond timer).

Table 5. Performance of processing steps time in RIVLPR

Image Preprocessing Steps	Average time (milliseconds)	
	MIDM Level 1	MIDM Level 3
Vehicle detection	150	170
Vehicle license plate detection	132	205
Vehicle license plate segmentation	122	134
Vehicle license plate processing	65	88
Character recognition	188	260
Reliability Rate of recognition	3	3
TOTAL TIME	660	860

Table 6. Performance of single level VLPR and MIDM

Model	Vehicle detection	Plate detection	Plate recognition
Single level	99%	97%	95.39%
Multilevel	100%	100%	98.68%

Table 7. Performance of Plate location detection and recognition

Vehicle	False detection	True Detection	Number Of Images	Rate
Detection	0	500	500	100%
Recognition	7	491	498	98.68%

Table 8. Performance of Plate Number Recognition

Recognition	Plate's characters	
	Numerals	Letters
False detection	10	0
True detection	3476	498
Number of Numeral/Letter	3486	498
Recognition rate	97.72%	100
TOTAL RECOGNITION	98.86%	

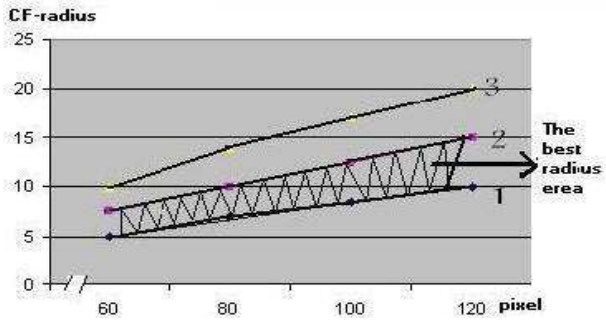

Figure 22. Different CF radius Between-the-Lights

At first, in vehicle detection section a true detection rate of 99.25% and a rejection rate of 1.47% were achieved. Table 6 depicts the processing time of the proposed system for every processing steps. Table 7 shows the performance of single level and multilevel model that indicated MDIM improved accuracy. As can be observed, Table 8 depicts the results of plate detection 100%; also the rate of plate recognition is above 98.86%. In Table 9, we compare the performance of vehicle license plate recognition in details. Finally, the inexistence plate recognition rate is 100%. False recognition with reliable analysis and true recognition with unreliable analysis error rates are 0.1% and 0.2%, respectively.

The major problems in the proposed algorithmic sequence revolve around the varying light levels encountered during a twenty-four hour period especially at night. The effect that lighting changes have on the image being forwarded to the POCR program is similar to the physically damaged appearance of the plates. This problem can be reduced by applying some new techniques and the MIDM in the two main parts (plate detection and recognition) of RIVLPR. Finally, Figure 22 shows the experimental results of different radius for finding Between-the-Lights in vehicle detection. The best radius area is between lines 1 and 2.

10. Conclusion

There are frequent situations in which a system's ability to recognize registration numbers can be useful. This paper presents such situations, a system designed to satisfy the requirements and some experimental results obtained with this system. The main features of the system presented are:

- Expandable designed with multithreading architecture for implementation on a parallel architecture and speed improvement.
- The plate and its number can be extracted accurately on a small size of images independent of color, size and position of its plate in a reasonable camera-vehicle distance.
- Vehicle plate number recognized with a self assessment alarm of the output reliability rate to its operator.
- Can work viably on both off-line and on-line, in outdoor as well as indoor environments and start with vehicle detection.

The system has been designed using an object oriented approach with multithreading architecture which allows easy upgrading and or the substitution of various sub-modules thus making it really implemented with a parallel processing technique and potentially suitable for a large range of vision applications. The performance of the system makes it a valid choice among its competitors especially in those situations where the system has to be used as an intelligent sensor. Furthermore, the modular architecture makes RIVLPR extremely flexible and versatile. A robust approach which considers different features of VLP to deal with more complex situations in the real world is presented. The proposed algorithms can be successfully run on our complex database. As shown in the experimental results section, the proposed algorithms are robust against different lighting conditions, shadows, small size images, and multi-plates vehicle.

This research has proven the capability of MIDM in all steps of Persian-Arabic VLP. In addition, some improvements would be beneficial to the development of the vehicle license plate recognition. We have shown that MIDM can be successfully used within the VLPR. The parallel processing capability (multithreading) presents the means for very fast image recognition. The spatially invariant character and the large amount of information that has to be analyzed make these parts obvious candidates for introduction to parallelism. The error rate is little. In the future, a better character recognition algorithm with rapid segmentation that can reduce the number of connected character regions must be found. To further improve on the speed of this system, ways to reduce error rate, and hence increase the speed of the algorithms toward a multinational plate recognition system (MPRS) with fully parallel architecture for scene analysis, and different message types for automatic correcting camera position or setting light imaging system must be investigated.

References

[1] A. Mahabadi, M. A. Torkamani, and M. Kazemian, "Recognition of Farsi Handwritten Digit using Fuzzy Logic," The CSI journal on Computer Science and Engineering, vol. 4, no. 1, pp. 19-25, Spring 2006.

[2] K. Tanabe, E. Marubayashi, H. Kawashima, T. Nakanishi, and A. Shio, "PC−Based Car license plate Reading," Image and Video Processing's, vol. SPIE−2182, pp. 220−231, 1994.

[3] S. Abe, and M. Lan, "A Classifier Using Fuzzy Rules Extracted Directly from Numerical Data," Proceedings of the Second IEEE International Conference on Fuzzy Systems, vol. 2, pp. 1191−1198, San Francisco, 1993.

[4] V. Shapiro, G. Gluhchev, and Dimov, "Towards a multinational car license plate recognition system," Machine Vision and Applications, vol. 17, no. 3, pp. 173–183, August 2006.

[5] C. Rahman, W. Badawy, and A. Radmanesh, "A real time vehicle's license plate recognition system," Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 163–166, 2003.

[6] A. Kuehnle, "Symmetry-based recognition for vehicle rears," Pattern Recognition Letters, vol. 12, pp. 249–258, 1991.

[7] M. Bertozzi, A. Broggi, A. Fascioli, and S. Nichele, "Stereo vision-based vehicle detection," IEEE Intelligent Vehicles Symposium, pp. 39–44, 2000.

[8] D. Guo, T. Fraichard, M. Xie, and C. Laugier, "Color modeling by spherical influence field in sensing driving environment," IEEE Intelligent Vehicle Symposium, pp. 249–254, 2000.

[9] H. Mori, and N. Charkai, "Shadow and rhythm as sign patterns of obstacle detection," International Symposium on industrial electronics, pp. 271– 277, 1993.

[10] E. Dickmanns, E. D. Behringer, R. Dickmanns, and D. Hildebrandt, "The seeing passenger car 'vamors-p," International Symposium on Intelligent Vehicles'94, pp. 24–26, 1994.

[11] C. Tzomakas, and W. Seelen, "Vehicle detection in traffic scenes using shadows," Technical Report 98-06, Institute fur neuroinformatik, Ruht-universitat, Bochum, Germany, 1998.

[12] M. Bertozzi, A. Broggi, and S. Castelluccio, "A real-time oriented system for vehicle detection," Journal of Systems Architecture, pp. 317–325, 1997.

[13] N. Matthews, P. An, D. Charnley, and C. Harris, "Vehicle detection and recognition in grayscale imagery," Control Engineering Practice, vol. 4, pp. 473–479, 1996.

[14] C. Goerick, N. Detlev, and M. Werner, "Artificial neural networks in real-time car detection and tracking applications," Pattern Recognition Letters, vol. 17, pp. 335–343, 1996.

[15] U. Handmann, T. Kalinke, C. Tzomakas, M. Werner, and W. Seelen, "An image processing system for driver assistance," vol. 18, no. 5, 2000.

[16] P. Parodi, and G. Piccioli, "A feature-based recognition scheme for traffic scenes," IEEE Intelligent Vehicles Symposium, pp. 229–234, 1995.

[17] Z. Sun, R. Miller, G. Bebis, and D. DiMeo, "A real-time precrash vehicle detection system," IEEE International Workshop on Application of Computer Vision, December 2002.

[18] T. Kalinke, C. Tzomakas, and W. V. Seelen, "A texture-based object detection and an adaptive model-based classification," IEEE International Conference on Intelligent Vehicles, pp. 143–148, 1998.

[19] R. Cucchiara, and M. Piccardi, "Vehicle detection under day and night illumination," International ICSC Symposium on Intelligent Industrial Automation, 1999.

[20] J. Hancock, "High-speed obstacle detection for automated highway applications," Technical Report RI-TR-97-17, Carnegie Mellon University, 1997.

[21] U. Franke, and I. Kutzbach, "Fast stereo based object detection for stop-go traffic," Intelligent Vehicles, pp. 339–344, 1996.

[22] H. Mallot, H. Bulthoff, J. Little, and S. Bohrer, "Inverse perspective mapping simplifies optical flow computation and obstacle detection," Biological Cybernetics, Vol. 64, No. 3, pp. 177–185, 1991.

[23] G. Zhao, and S. Yuta, "Obstacle detection by vision system for autonomous vehicle," Intelligent Vehicles, pp. 31–36, 1993.

[24] M. Bertozzi, and A. Broggi, "Gold: A parallel real-time stereo vision system for generic obstacle and lane detection," IEEE Transaction on Image Processing, vol. 7, pp. 62–81, 1998.

[25] T. Williamson, and C. Thorpe, "A trinocular stereo system for highway obstacle detection," IEEE International Conference on Robotics and Automation, 1999.

[26] A. Giachetti, M. Campani, and V. Torre, "The use of optical flow for road navigation," IEEE transactions on robotics and automation, vol. 14, no.1, pp. 34–48, 1998.

[27] J. Weng, N. Ahuja, and T. Huang, "Matching two perspective views," IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 14, pp. 806–825, 1992.

[28] D. Koller, N. Heinze, and H. Nagel, "Algorithmic characterization of vehicle trajectories from image sequence by motion verbs," IEEE International Conference on Computer Vision and Pattern Recognition, pp. 90–95, 1991.

[29] B. Heisele, and W. Ritter, "Obstacle detection based on color blob flow," IEEE Intelligent Vehicle Symposium, pp. 282–286, 1995.

[30] T. Ito, K. Yamada, and K. Nishioka, "Understanding driving situations using a network model," Intelligent Vehicles, pp. 48–53, 1995.

[31] A. Bensrhair, M. Bertozzi, A. Broggi, P. Miche, S. Mousset, and G. Moulminet, "A cooperative approach to vision-based vehicle detection," IEEE Intelligent Transportation Systems, pp. 209–214, 2001.

[32] H. Schneiderman, and T. Kanade, "A statistical method for 3d object detection applied to faces and cars," IEEE International Conference on Computer Vision and Pattern Recognition, pp. 746–751, 2000.

[33] S. Kawato, and N. Tetsutani, "Real-time Detection of Between-the-Eyes with a Circle Frequency Filter," The 5th Asian Conference on Computer Vision, Melbourne, Australia, pp. 141-146, January 2002.

[34] F. Martín, M. García, and J. L. Alba, "New Methods for Automatic Reading of VLP's (Vehicle License Plates)," IASTED International Conference on Signal Processing, Pattern Recognition and Applications, SPPRA 2002. [Online]. Avalable: http://www.gpi.tsc.uvigo.es/pub/papers/sppra02.pdf.

[35] B. Hongliang, and L. Changping, "A hybrid License Plate Extraction Method Based on Edge Statistics and Morphology," Proceedings of Conference on Pattern Recognition (ICPR), pp. 831-834, 2004.

[36] T. H. Wang, F. C. Ni, K. T. Li, and Y. P. Chen, "Robust License Plate Recognition based on Dynamic Projection Warping," proceedings of IEEE International Conference on Networking, Sensing & Control, pp.784-788, 2004.

[37] S. Xifan, Z. Weizhong, and S. Yonghang, "Automatic License Plate Recognition System Based on Color Image Processing," Lecture Notes on Computer Science (LNCS) 3483, O. Gervasi et al. (Eds) Springer-Verlag, pp. 1159 – 1168, 2005.

[38] D. Yan, M. Hongqing, L. u. Jilin, and L. Langang, "A High Performance License Plate Recognition System Based on the Web Technique," Proceedings of Conference on Intelligent Transportation Systems, pp. 325-329, 2001.

[39] P. Comelli, P. Ferragina, M. N. Granieri, and F. Stabile, "Optical recognition of motor vehicle license plates," IEEE Transaction On Vehicular Technology, vol. 44, no. 4, pp. 790-799, 1995.

[40] S. Draghici, "A neural network based artificial vision system for license plate recognition, International Journal of Neural Systems," vol. 8, no. 1, pp. 113-126, 1997.

[41] A. Broumandnia, and M. Fathy, "Application of pattern recognition for Farsi license plate recognition," ICGST-CVIP Journal, vol. 5, no. 2, pp. 25-31, 2005.

[42] N. Zimic, J. Ficzko, M. Mraz, and J. Virant, "The Fuzzy Logic Approach to the Car Number Plate Locating Problem," proceedings of Intelligent Information Systems (IIS'97), pp. 227-230, 1997.

[43] J. A. G. Nijhuis, M. H. 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," proceedings of IEEE International Conference on Neural Networks, vol. 5, pp. 2232-2236, 1995.

[44] S. L. Chang, L. S. Chen, Y. C. Chung, and S. W. Chen, "Automatic License Plate Recognition," IEEE Transaction on Intelligent Transportation Systems, vol. 5, no. 1, pp. 42-53, 2004.

[45] G. Adorni, S. Cagnoni, M. Gori, and M. Mordonini, "Access control system with neuron-fuzzy supervision," proceedings Conference on Intelligent Transportation Systems, pp. 472-477, 2001.

[46] T. D. Duan, T. L. Hong Du, T. V. Phuoc, and N. V. Hoang, "Building an Automatic Vehicle License Plate Recognition System," proceedings of International Conference in Computer Science (RIVF), pp. 59-63, 2005.

[47] L. Dlagnekov, "License Plate Detection Using AdaBoost," Computer Science & Engineering Department, San Diego, La Jolla, 2004, [Online]. Available: http://www.cse.ucsd.edu/classes/ fa04/cse252c/projects/louka.pdf.

[48] C. T. Hsieh, Y. S. Juan, and K. M. Hung, "Multiple License Plate Detection for Complex Background," Proceedings of International Conference on Advanced Information Networking and Applications (AINA), vol. 2, pp. 389-392, 2005.

[49] C. N. Anagnostopoulos, I. Anagnostopoulos, V. Loumos, and E. Kayafas, "A license plate recognition algorithm for Intelligent Transportation System applications," Proceedings of the Third IEEE Transactions on T-ITS-05-08-0094, vol. 1, pp. 1-16, May 2008.

[50] D. S. Kim, and S. I. Chien, "automatic Car License Plate Extraction Using Modified Generalized Symmetry Transform and Image Warping," proceedings of International Symposium on Industrial Electronics (ISIE), pp. 2022-2027, 2001.

[51] M. H. Brugge, J. H. Stevens, J. A. G. Nijhuis, and L. Spaanenburg, "License Plate Recognition Using DTCNNs," proceedings of IEEE international Workshop on Cellular Neural Networks and their Applications, pp. 212-217, 1998.

[52] I. Mario, M. Chacon, and S. A. Zimmerman, "License Plate Location Based on a Dynamic PCNN Scheme," proceedings of International Joint Conference on Neural Networks, vol. 2, pp. 1195 – 1200, 2003.

[53] K. K. Kim, K. I. Kim, J. B. Kim, and H. J. Kim, "Learning-Based Approach for License Plate Recognition," proceedings of IEEE Signal Processing Society Workshop, Neural Networks for Signal Processing, vol. 2, pp. 614 - 623, 2000.

[54] N. R. Pal, and S. K. Pal, "A review on image segmentation techniques, Pattern Recognition," vol. 26, no. 9, pp. 1277–1294, 1993.

[55] D. A. forsyth, J. ponce, Computer Vision: A Modern Approach, New Jersey: Prentice Hall, 2003.

[56] S. Nomura, K. Yamanaka, O. Katai, H. Kawakami, and T. Shiose, "A novel adaptive morphological approach for degraded character image segmentation," Pattern Recognition, vol. 38, vo. 11, pp. 1961-1975, November 2005.

[57] Y. Cui, and Q. Huang, "Extracting characters of license plates from video sequences," Machine Vision and Applications, Vol. 10, pp. 308-320, 1998.

[58] C. N. Anagnostopoulos, I. Anagnostopoulos, V. Loumos, and E. Kayafas. "A license plate recognition algorithm for Intelligent Transportation System applications," Proceedings of the Third IEEE Transactions on T-ITS-05-08-0094, vol. 1, pp. 1-16, May 2008.

[59] B. R. Lee, K. Park, H. Kang, H. Kim, and C. Kim, "Adaptive Local Binarization Method for Recognition of Vehicle License Plates," Lecture Notes on Computer Science 3322 R. Klette and J. Žunić (Eds.), Springer-Verlag, pp. 646–655, 2004.

[60] P. K. Sahoo, S. Soltani, A. K. C. Wong, and Y. C. Chen, "A survey of thresholding techniques," Computer Vision Graphics Image Processing, vol. 41, pp. 233–260, 1988.

[61] T. Naito, T. Tsukada, K. Yamada, K. Kozuka, and S. Yamamoto, "Robust license-plate recognition method for passing vehicles under outside environment," IEEE Transaction on Vehicular Technology, vol. 49, no. 6, pp. 2309-2319, 2000.

[62] D. Llorens, A. Marzal, V. Palazon, and J. M. Vilar, "Car License Plates Extraction and Recognition Based on Connected Components Analysis and HMM Decoding," Lecture Notes on Computer Science 3522, J. S. Marques (Eds) Springer-Verlag, pp. 571–578, 2005.

[63] C. Anagnostopoulos, E. Kayafas, and V. Loumos, "Digital image processing and neural networks for vehicle license plate identification," Journal of Electrical Engineering, vol. 1, no.2, pp. 2-7, 2000. [Online]. Avalable: http://www.medialab.ntua.gr/people/canag/journals.php.

[64] C. Anagnostopoulos, T. Alexandropoulos, S. Boutas, V. Loumos, and E. Kayafas, "A template-guided approach to vehicle surveillance and access control," proceedings of IEEE Conference on Advanced Video and Signal Based Surveillance, pp. 534 – 539, 2005.

[65] Y. Hu, F. Zhu, and X. Zhang, "A Novel Approach for License Plate Recognition Using Subspace Projection and Probabilistic Neural Network," Lecture Notes on Computer Science 3497, J. Wang, X. Liao, and Z. Yi, Ed. Springer-Verlag, LNCS 3497, pp. 216–221, 2005.

[66] T. Sirithinaphong, and K. Chammnongthai, "The recognition of car license plate for automatic parking system," Proceedings of 5th International Symposium on Signal Processing and its Application, pp. 455-457, 1998.

[67] G. Luo, "Fast wavelet image denoising based on local variance and edge analysis," international journal of intelligent technology, vol. 1, no. 2, pp. 165-175, 2006.

[68] M. Sezgin, and B. Sankar, "Survey over image thresholding techniques a quantitative performance evaluation," journal of electronic imaging, vol. 13, no. 1, pp. 146-165, 2004.

[69] M. Okutomi, and T. Kanade, "A multiple-baseline stereo," IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 15, pp. 353–363, 1993.

[70] F. Chang, K. H. Liang, T. M. Tan, and W. L. Hwang, "Binarization of Document Images Using Hadamard Multiresolution Analysis," Fifth International Conference on Document Analysis and Recognition (ICDAR'99), ICDAR, pp. 157-161, 1999.

[71] K. Fukui, and O. Yamaguchi, "Facial feature point extraction method based on combination of shape extraction and pattern matching," Systems and Computers in Japan, vol. 29, no. 6, pp. 49–58, 1998.

[72] J. Yang, R. Stiefelhagen, U. Meier, and A. Waibel, "Real-time face and facial feature tracking and applications," Proceedings of Workshop on Audio-Visual Speech Processing, pp. 79–84, 1998.

[73] J. Arvo, M. Hirvikorpi, and J. Tyystjarvi, "Approximate Soft Shadows with an Image-Space Flood-Fill Algorithm," EUROGRAPHICS, vol. 23, no. 3, pp. 271-280, 2004.

[74] GA666-2006. People's Republic of China regulations for vehicles license plate (in Chinese), Nov. 2006.

[75] Web page of http://thefreeddictionary.com for number plates regulations of many countries.

Aminollah Mahabadi received his B.Sc. degree in Electrical Engineering from Iran Science and Technology University in Iran, in 1990, and M.Sc. degree in Computer Engineering from Amirkabir University in Iran, in 1996. He is the author of five published books. His research interests are in ITS, image processing, scheduling, Network on Chip, and simulation. He currently instructs in the Department of Electrical and Engineering at Shahed University, Iran.

E-mail: mahabadi@shahed.ac.ir

Akbar Ranjbar was born in Tehran, Islamic Republic of Iran. He received his B.S. from the University of Arkansas at Fayetteville in 1985, and his M.S. from University of New Orleans in Louisiana in 1988 in USA. He is currently a member of faculty at Shahed University, Engineering College, Electrical Engineering Department, Electronics

Group. He was one of the founders of Engineering College at Shahed University which was established in 1992. He is also working at Acoustics Laboratories at Shahed University and Space Research Institute. Since 2000 his research has included satellite sensors, acoustics sensors and detection devices, and acoustic signal processing with specific interest in instrumentations.

E-mail: ranjbar@shahed.ac.ir

Paper Handling Data:

Submitted: 23.08.2009
Received in revised form: 06.10.2012
Accepted: 09.11.2012
Corresponding author: Aminollah Mahabadi,
Department of Electrical Engineering, Shahed University, Tehran, Iran.