TELKOMNIKA Indonesian Journal of Electrical Engineering
Vol.12, No.1, January 2014, pp. 584 ~ 591
DOI: http://dx.doi.org/10.11591/telkomnika.v12i1.3097

Rectification of License Plate Images Based on Hough Transformation and Projection

Hongyao Deng*[1, 2], Qingxin Zhu[1], Jingsong Tao[3], Hao Feng[4]
[1]School of Information & Software Engineering, University of Electronic Science and Technology of China, No. 2006, Xiyuan Ave, West Hi-Tech Zone, Chengdu, 611731, Sichuan China, Ph/Fax: +86 28 83201864
[2]School of Mathematics & Computer Science, Yangtze Normal University, No. 98 Julong Rd., Lidu Fuling District, Chongqing, 408000, China. Ph: +86 23 07290088
[3]School of Electrical Engineering, Wuhan University, Luojiashan, Wuchang, Wuhan, 430072, China, Ph: +86 27 68770776
[4]School of Computer Science and Engineering, Guilin University Of Electronic Technology (GUET), No.1 Jinji Road, Guilin, 541004, Guangxi China, Ph/Fax: +86-773 5896528
Corresponding author, e-mail: hydeng_2004@163.com[1,2], qxzhu@uestc.edu.cn[1], jamson_tao@163.com[3], fengh@guet.edu.cn[4]

Abstract

It is crucial to segment characters correctly and improve rate of correct character recognition when processing automobile license plates corrections. In this paper, two algorithms are proposed to obtain the horizontal tilt and vertical shear angles. The transformation matrix for images rectification is given and the subpixel issue is solved. Some experiments were done to test the algorithms. Experimental results show that the algorithm is robust, flexible and effective.

Keywords: *license plate; horizontal tilt; vertical shear; subpixel issue; plate correction*

1. Introduction

LPR (License Plates Recognition) is a crucial element for implementation of the ITS (Intelligent Transportation System). Generally, a typical algorithm for the LPR technique consists of three major parts: license plate location, license plate character segmentation and license plate character identification. The conventional approaches to locating license plates attempt to find the license plate by means of license plates image features such as the color, shape, symmetry, geometry or the gradient [1, 2]. For example, in our earlier work, license plate regions were extracted by both the dynamic RGB (which are the three color channels: Red, Green and Blue) threshold formula and the gradient of pixel intensities over the horizontal rows [3]. The next task is license character segmentation. However, the license plates extracted from images are not usually rectangular due to perspective or other deformations and extremely disadvantageous for segmentation of license plate characters. Therefore, it is necessary to rectify the license plate image before characters are segmented.

To correct license plates, there are two parameters that play important roles. One is the horizontal tilt; another is the vertical tilt, also called shear angle. At present many correction algorithms have been proposed. Literature [4] proposed differential projection algorithm to correct the horizontal tilt (the author's representation). In fact, it is a horizontal differential histogram. The principal method sums up the different intensities of all immediate adjacent pixels when an angle is obtained by rotating the license plate image. The angle corresponding closely to the maximum is the horizontal tilt angle. It is evident that the algorithm is considerably sensitive to noise. [5, 6] checked the upper and lower lines on the license plate image using only Hough transformation. They considered a thing justified when the upper line is always parallel with the lower line without respect to the deformation of the license plate. There is an error made in [5]. That is, that shear correction was used to correct license plate regions by the angle only obtained through Hough Transformation. Literature [6] used the Hough transform to determine the four corner coordinates of a plate region and corrected the distorted images through a bilinear interpolation transformation. For this algorithm, it is very difficult to find the actual vertex coordinates if image quality is not very high. There is another method that can be

Received May 4, 2013; Revised July 23, 2013; Accepted August 7, 2013

seen in Literature [7, 8]. The straight line fitting method was used to correct the horizontal tilt of license plate, and the horizontal projection method was used to correct the vertical tilt by finding the minimum variance. The drawback of this approach to license plate correction is that it is sensitive to noise. To overcome the issue, the author chose a suitable de-noising arithmetic before correction. In literature [9], a robust algorithm was proposed to correct license plates. It obtained the corrected parameters though the geometric configuration between two neighboring numeric character blobs. Of course, there are many other methods to cope with the plate correction issue. Here no more than mention of key exemplars of these approaches.

The focus of this paper is how to obtain the two type parameters to correct license plates. It is organized as follows. The next section presents the Hough Transformation, and proposes an algorithm by which the horizontal correction parameter was obtained. In section 3, a projection algorithm is proposed to get the shear parameter. The correction transformation formula is described in Section 4. Additionally, the subfixels issue is solved in this section. Experimental results and conclusion are presented in Section 5.

2. Obtaining the Tilt Correction Parameter

License plates require horizontal tilt correction, which can be executed by means of the horizontal tilt angle. Major approaches check the edges of license plates. So far, there are many typical algorithms are proposed such as Sobel Edge Detector, Prewitt Edge Detector, Roberts Edge Detector, Laplacian of Gaussian (LoG) Detector, Zero-Crossings Detector, and Canny Edge Detector [10-12]. These algorithms sometimes cannot check the edge correctly due to the influence of uneven illumination, noise and deformation. The standard Hough Transformation is a method to check straight lines. This line detection algorithm does not limit performance, even with noise and fragmentation. In this paper, we checked the upper and lower lines on license plates by means of the Hough Transformation, and found the character blobs between two lines. The horizontal tilt angle was obtained by the two lines and geometric configuration features of character blobs.

2.1. Hough Transformation

The principle of Hough Transformation can be seen from the literature [13, 14]. Give a set of points in an image (typically a binary image), with the Hough Transformation, we consider a point (x_i, y_i) and all the lines that pass through it. Infinitely many lines pass through (x_i, y_i), all of which satisfy the slope-intercept equation $y_i = ax_i + b$ for some values of a and b. Writing this equation as $b = -x_i a + y_i$ and considering the *ab*-plane (also called parameter space) yields the equation of a *single* line for a fixed pair (x_i, y_i). Furthermore, a second point (x_j, y_j) also has a line in parameter space associated with it, and this line intersects the line associated with (x_i, y_i) at (a', b'), where a' is the slope and b' the intercept of the line containing both (x_i, y_i) and (x_j, y_j) in the *xy*-plane. In fact, all points contained on this line have lines in parameter space that intersect at (a', b'). Figure 1(a) illustrates these concepts.

In principle, the parameter-space lines corresponding to all image points (x_i, y_i) can be plotted, and then image lines could be identified by where large numbers of parameter-space lines intersect. A practical difficulty with this approach, however, is that *a* (the slope of the line) approaches infinity as the line approaches the vertical direction. One way around this difficulty is to use the normal representation of a line:

$$x\cos\theta + y\sin\theta = \rho \tag{1}$$

Figure 1(b) illustrates the geometric interpretation of the parameters ρ and θ, and it represents the family of lines that pass through a particular point (x_i, y_i). The intersection point (ρ', θ') corresponds to the line that passes through both (x_i, y_i) and (x_j, y_j).

To detect the lines in image via Hough Transformation, the $\rho\theta$-parameter space is divided into so-called *accumulator cells*, as illustrated in Figure 1(c), where $(\rho_{\min}, \rho_{\max})$ and $(\theta_{\min}, \theta_{\max})$ are expected ranges of the parameter values. Usually, the maximum range of values is

$-90° \le \theta \le +90°$ and $-D \le \rho \le +D$, where D is the distance between corners in the image. The cell at coordinates (i, j), with accumulator value $A(i, j)$, corresponds to the square associated with parameter space coordinates (ρ_i, θ_j). Initially, these cells are set to zero. Then, for every non-background point (x_k, y_k) in the image plane, we let θ equal each of the allowed subdivision values on the θ axis and solve for the corresponding ρ using the equation $\rho = x_k \cos\theta + y_k \sin\theta$. The resulting ρ*-values* are then rounded off to the nearest allowed cell value along the ρ*-axis*. The corresponding accumulator cell is then incremented. At the end of this procedure, a value of Q in $A(i, j)$, means that Q points in the xy*-plane* lie on the line $\rho_j = x\cos\theta_j + y\sin\theta_j$. The parameter θ corresponding to the $A(i, j)$, in which the value is more than others, is the inclination angle by which we can correct the tilt of plate images.

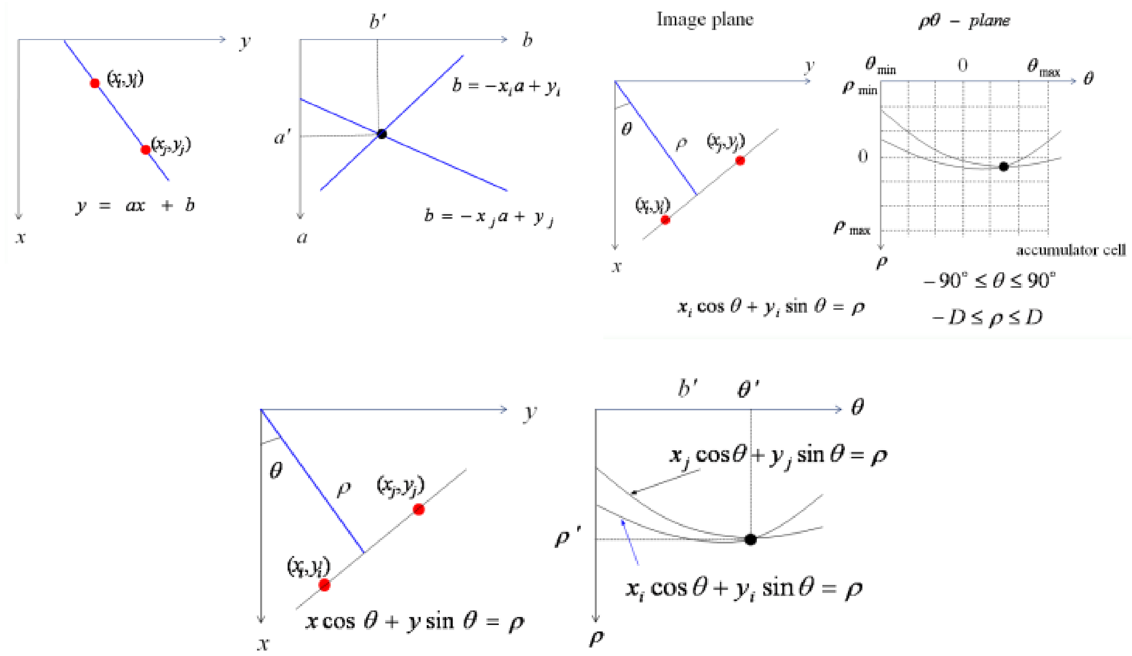


Figure 1. (a) slope-intercept equation; (b) normal representation; (c) Detecting lines via the HT

2.2. Computing Horizontal Tilt Angle

The two straight lines on the license plate region were detected via the Hough Transformation Algorithm. Let the average value of their horizontal tilt angles be θ_n. We searched the character blobs between the lines. In fact, they are not character blobs but some connected domain areas because of the fragmented, overlapping and connected characters. In our earlier work [15], we proposed an algorithm to solve the issue, and obtained normal character blobs. Let the successive character blobs be $S = \{s_1, \cdots, s_n\}$, then the minimum contain rectangle of each character blob(MCR of character blob) can be located. Suppose that the horizontal distance between the center positions of two neighboring character blobs are $OD = \{OD_i \mid i = 1, \cdots n-1\}$, and the vertical distance are $OH = \{OH_i \mid i = 1, \cdots n-1\}$ (Figure 2). A set of angles is then given by

$$\theta = \{\theta_i \mid \theta_i = \arctan(OH_i / OD_i), i = 1, \cdots, n-1\} \quad (2)$$

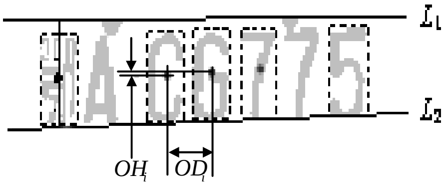


Figure 2. The geometrics configuration between neighboring characters.

The final horizontal tilt angle is

$$\theta^* = \frac{1}{n}\sum_{i=1}^{n}\theta_i \tag{3}$$

Above all, the algorithm can be seen in Table 1.

3. Shear Angle Parameter

As we saw earlier, the deformation of License plate not only has the tilt on the horizon but also vertical shear. The focus of the section is on the shear issue. Commonly, shear consists of X axis and Y axis direction deformation. The deformation of the shear mainly comes from the vertical direction after the tilt correction of license plates. The other deformation hardly affects license plates. Therefore, we shall not consider them.

Assume that the shear angle ψ is different from –φ to +φ, ψ∈[–φ, +φ]. The shear angle is just the vertical shear. To obtain the shear angle, we project the MCR of character blobs in some direction. Let the MCR of character blobs be $s_N = \{s_i \mid i = 1,\cdots n\}$ the length of projection be $L\left(S_i^{\psi}\right)$ (Figure 3), then we define a decision function as

$$L(\psi) = \sum_{Si\in\{S_N\}} L(S_i^{\psi}) \tag{4}$$

where ψ∈[-φ, +φ].

The function *L*(ψ) can convey the quality of shear angles, and the minimum of the *L*(ψ) is the optimal. To optimize the algorithm, a flag was set. If the projection is overlapping, the flag is false, otherwise it is true. In the iterative process of calculation, the calculation in its direction will be abandoned if the flag is false, because it is not the best optimal, and will choose another angle to process. Let the $\Delta\psi$ be the iterative step size, which determines the accuracy of the shear angle. The algorithm is described in Table 2.

4. License Plate Rectification

4.1. Rectification Transformation

The horizontal tilt correction parameter θ^* and shear parameter ψ^* were obtained in the third and fourth sections. In this section, the two parameters will be used to rectify the license plate image. The transformation matrix of horizontal tilt correction and shear correction were defined respectively as

$$R(\theta^*) = \begin{bmatrix} \cos\theta^* & -\sin\theta^* & 0 \\ \sin\theta^* & \cos\theta^* & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad SH(\psi^*) = \begin{bmatrix} 1 & 0 & 0 \\ \tan\psi^* & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

where the θ^* and ψ^* are the horizontal tilt correction and shear parameters.

Table 1. Obtaining the horizontal tilt angle

/*Algorithm 1: Computing the tilt correction parameter*/

1. Eliminate the points on the background image, image binaryzation, create the image space $I[X,Y]$;
2. //check the line via HT algorithm.
3. Define a parameter space $Hu[\rho,\theta]$, initial it with 0;
4. for each point $I_i(x,y) \in I$ { //searching all points on the image
5. for($\theta = 0; \theta < 360; \theta++$){
6. Computing $\rho = x\cos\theta + y\sin\theta$;
7. $Hu(\rho_i,\theta_i) \leftarrow Hu(\rho_i,\theta_i)+1$; // if hit a cell, then its accumulator increment by 1.
8. }
9. }
10. $Hu(\rho',\theta') \leftarrow \max\{Hu\}$;
11. Computing the θ_n , and define new image space $I^{'}[X,Y]$; //The region between the two lines.
12. //Searching the connected domains.
13. $I^{'}[X,Y]$ round its outline extended 1 pixel width with 0; //The purpose is to unify the processing method with 8-unit-neiboring scheme.
14. Label all the destination points with no tag and define a set of MCR blobs, $S[n] = \{x_1, y_1, x_2, y_2\}$; $k \leftarrow 0$;
15. repeat: searching the points on $I^{'}$
16. Find a destination point, $p_i(x,y)$;
17. if($p_i(x,y)$ has no tag){ //take the point *p* as seed.
18. $S[k] \leftarrow s(x,x,y,y)$; $p(x,y) \leftarrow p_i(x,y)$; //Initiate the coordinates of MCR
19. repeat //Obtain a MCR
20. Push the destination points of 8-unit-neiboring of $p(x,y)$ to stack;
21. Pop a point from the top of the stack{
22. Label the point $p(x_i, y_i)$ with tag;
23. Adjust the coordinates of the MCR{
24. if $(x_i < x_1)$ then $x_1 \leftarrow x_i$; if $(y_i < y_1)$ then $x_1 \leftarrow x_i$;
25. if $(x_i > x_2)$ then $x_2 \leftarrow x_i$; if $(y_i > y_2)$ then $y_2 \leftarrow y_i$;
26. }
27. $p(x,y) \leftarrow p(x_i, y_i)$; goto row 20;
28. }
29. until(the stack is nothing);
30. }
31. until(all the points on $I^{'}$ are searched)
32. Computing the $\{\theta_i \mid i = 1,\cdots,n-1\}$ by the MCR blobs;
33. return $\frac{1}{n}\sum_{i=1}^{n}\theta_i$; //return the horizontal tilt angle,

Note: In a MCR, $S(x_1, y_1, x_2, y_2)$, (x_1, y_1, x_2, y_2) notates the upper left and the lower right coordinates.

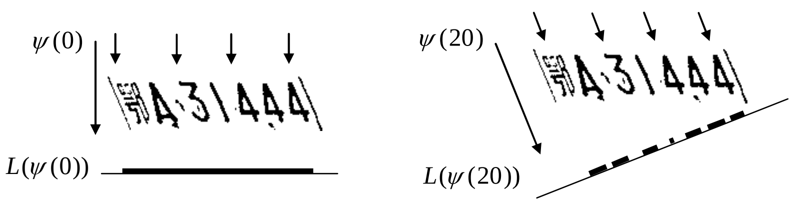


Figure 3. The different values of the projection function in different directions.

Table 2. Obtaining the vertical shear angle.

/*Algorithm 2: Computing the vertical tilt angle correction parameter*/	
1.	Define the step size Δψ; Let a projection angle be ψ=-φ+Δψ, candidate optimal shear angle be ψ^*=0, a projection function of MCR be L(ψ), candidate shortest projection function be $L^*(\psi^*)$;
2.	repeat
3.	adjust projection direction, ψ←(ψ+Δψ);
4.	Project each character MCR {S_0,…, S_n} in direction ψ{
5.	if(projection overlap)
6.	goto (3);
7.	else
8.	computer L(ψ)←$\Sigma_{si\in Sn}$L(S_i);
9.	}
10.	if(L(ψ)< $L^*(\psi^*)$)
11.	$L^*(\psi^*)$←L(ψ);
12.	until(ψ>φ)
13.	return(ψ^*)

The final correction transformation is

$$(x', y', 1)^T = A(x, y, 1)^T \quad (5)$$

where $A=R(\theta^*)*SH(\psi^*)$, (x,y) is the point on the original image and (x',y') is the point after transformation.

4.2. Decimal Fraction Issue

Obviously, the pixel coordinates (x', y') obtained by equation (5) may be decimal. We call them subpixel. However, the coordinates of digital images must be integer values. If the pixel coordinate (x', y') is decimal, it should be retained because it is also the point on the image. This issue is addressed as follows.

Assume that the coordinate's values of the point $p(x', y')$ are decimal. And its 4-unit-neighboring points of which are $p_i(i = 0,1,2,3)$. The square was divided into 4 rectangles $R_i(i = 0,1,2,3)$ by the subpixel point p (Figure 4). Let the area of the R_i is $A_i(i = 0,1,2,3)$ respectively, then, the sum of them is 1, $\sum_{i=0}^{3} A_i = 1$. Assume that the probability that the p hits the square (not containing the four corner points) is equal probability. We can easily obtain that the $p(R_i) = A_i$ and $\sum_{i=0}^{3} p(R_i)$ =1. Let the $f(\bullet)$ be the intensity of the pixel. Note that the image is not the black-and-white image, but a gray image. Then the decimal fraction issue can be coped with equation (6).

$$f(p_i') = p(A_i) * f(p) + f(p_i) \quad (6)$$

where $f(p_i)$ notate the intensity before processing, $f(p_i^{'})$ notate the intensity after processing, $p(A_i)$ is the probability that the point p hits the rectangle R_i.

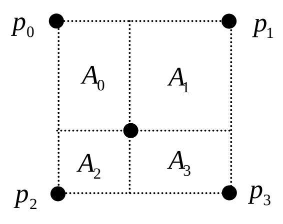


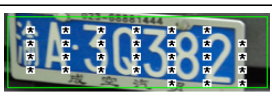
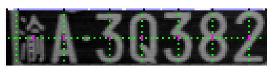
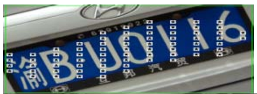
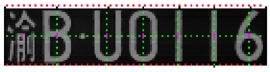
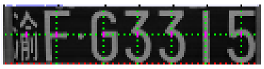
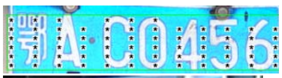
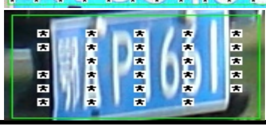
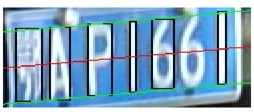
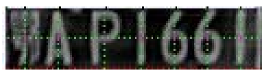
Figure 4. The geometrics configuration of subpixel

5. Experiment and Conclusion

To evaluate the flexibility and effectiveness, more than 200 images, shot randomly under various conditions, were tested. The size of the images was 640 × 480 pixels, and the format was 24-bit BMP or JPG. The license plates extraction were finished in our earlier work [3]. The next tasks are character segmentation [15] and recognition. The techniques in this paper preprocess the image before character segmentation, the purpose is to improve the condition of character segmentation. As was discussed previously, it consists of the horizontal tilt and vertical shear transformation. The typical experimental results were shown in Table 3.

As far as the complexity concerned, it is O(n^2) for the entire algorithms, coming primarily from the HT and MCR algorithms. However, the quantity of processing data is small for a license plate. The algorithm was optimized by cutting the hard data set (See Algorithm 2). The running time is not more than one millisecond on the VC6.0 platform in the environment of the XP OS and 2M memory. Above all, the conclusion can be drawn that we developed a simple and novel algorithm to rectify license plates, and the algorithm is characteristically robust, flexible, and effective.

Table 3. The typical experimental correction results

	The license plate extracted from the images.	The Hough transformation and the geometrics configuration of characters	The horizontal tilt correction	The vertical shear correction
1				
2				
3				
4				
5				
6				

Aknowledagement

This work is partially surpported by Visiting Scholarship of State Key Laboratory of Power Transmission Equipment & System Security and New Technology (Chongqing University) under grant No. 2007DA10512711412.

References

[1] Christos Nikolaos E, Anagnostopoulos, Ioannis E. A License Plate-Recognition Algorithm for Intelligent Transportation System Applications. *IEEE Transactions on Intelligent Transportation System.* 2006; 7(3): 371-391.

[2] Guangrong Ji, Hongjie Yi. *The Recognition of the Vehicle License Plate Based on Modular Networks*. Proceedings, Part I. International Symposium on Neural Networks. Dalian, China. 2004; 3173: 1015-1020.

[3] Deng Hongyao; Song Xiuli. *A novel approach for license plate location in natural images.* Applications of Digital Information and Web Technologies. London. 2009: 563-568.

[4] Wu Chengdong, Fan Yuquan. License Plate Character Segmentation Based on Differencing Projection and Preferably Segmented Character. *Journal of Northeastern University* (Natural Science). 2008; 29(7): 920-923.
[5] Zhou Kaijun, Chen Sanbao. Research of Vehicle License Plate Location and Character Segmentation Under Complex Scenes. *Computer Engineering*. 2007; 33(4): 198-200.
[6] Chen Zhen-Xue, Wang Guo-You. A Novel Algorithm of Character Segmentation in License Plates Recognition. *Microelectronics & Computer.* 24(2): 42-44.
[7] Kaushik Deb, Andrey Vavilin. *License Plate Tilt Correction Based on the Straight Line Fitting Method and Projection*. ICCCI 2010, Part III, LNAI 6423. 2010: 184-193.
[8] Kaushik Deb, My Ha Le. *Automatic Vehicle Identification by Plate Recognition for Intelligent Transportation System Applications*. IEA/AIE 2011, Part II, LNAI 6704. 2011: 163-172.
[9] DJ Kang. Dynamic Programming-based Method for Extraction of License Plate Numbers of Speeding Vehicles on the Highway. *International Journal of Automotive Technology.* 2009; 10(2): 205-210.
[10] Wei Jiang, Kin-Man Lam. Efficient Edge Detection Using Simplified Gabor Wavelets. *IEEE Transactions on Systems, Man and Cybernetics—Part B: Cybernetics.* 2009; 39(4): 1036-1047.
[11] Bai Hongliang, Liu Changping. *A hybrid License Plate Extraction Method Based On Edge Statistics and Morphology*. Pattern Recognition, ICPR 2004. Proceedings of the 17th International Conference. 2004: 23-26.
[12] Vahid Abolghasemi, Alireza Ahmadyfard. An edge-based color-aided method for license plate detection Image and Vision Computing. 2009; 27(8): 1134-1142.
[13] Chutatape O, Linfeng Guo. A modified Hough transform for line detection and its performance. *Pattern recognition*. 1999; 32(2): 181-192.
[14] Kamat, Varsha. *An efficient implementation of the Hough transform for detecting vehicle license plates using DSP'S*. Real-Time Technology and Applications Symposium.1995: 58-59.
[15] Deng Hongyao, Song Xiuli. *License Plate Characters Segmentation Using Projection and Template Matching*. International Conference on Information Technology and Computer Science. wuhan China. 2009; 1: 534-537.