EE 5359 MULTIMEDIA PROCESSING

Vehicle License Plate Detection Algorithm Based on Statistical Characteristics in HSI Color Model

Under the guidance of Dr. K. R. Rao

Submitted by: Prasanna Venkatesh Palani

UTA ID: 1000660520
Contents:

1. Acknowledgements .............................................................................................................3
2. List of acronyms .....................................................................................................................4
3. List of figures ..........................................................................................................................5
4. Abstract .................................................................................................................................6
5. Introduction ............................................................................................................................7
6. Vehicle License Plate Detection Module ..............................................................................8
   6.1 Color Segmentation ............................................................................................................9
   6.2 RGB Color Model ............................................................................................................9
   6.3 HSI Color Model .............................................................................................................10
   6.4 Morphological Processing ..............................................................................................14
   6.5 Edge Detection, Labeling, Filtering ...............................................................................15
   6.6 Character Recognition in Vehicle License Plate .............................................................18
7. Experimental results .............................................................................................................20
8. Conclusion ..............................................................................................................................22
9. Future work ............................................................................................................................22

References ..................................................................................................................................23
1. Acknowledgement

I would sincerely like to thank Dr. K.R.Rao for his constant guidance, support and motivation which led to the successful completion of this project.

I would also like to thank all my friends and the entire MPL lab members for their support and encouragement.
2. List of acronyms:

- VPLR - Vehicle License Plate Recognition
- RGB - Red, Green, Blue
- HSI - Hue, Saturation, Intensity
- MATLAB – Matrix Laboratory
- ITS – Intelligent Transport System
- VPLD - Vehicle License Plate Recognition
3. List of figures:

6.1. Algorithm of Implementation for vehicle license plate detection.

6.2. RGB Color Model.

6.3. HSI Color Model.

6.4. RGB Car Image.

6.5. HSI Car Image.


6.7. Morphological Processing.

6.8. Edge Detection.


6.10. Candidate Region.

6.11. Region of Interest.

6.12. Character Extraction

6.13. Alphanumeric Templates

6.14. Plot of 2 D Cross Correlation of one character recognized with 36 templates
4. ABSTRACT:

Image capturing and processing has become an important part of intelligent robotics. It is mainly used to detect an object in real-time. With constantly increasing traffic on roads, there is a need for intelligent traffic management system. License plate detection is widely used for detecting speeding cars, security control, traffic law enforcement and electronic toll collection. License plate detection can be performed via various approaches such as vector quantization, Gabor transform, optical character recognition, neural networks etc. Character recognition in license plate is an inevitable component of vehicle license plate recognition (VLPR) system. Lot of research has been carried out to detect the characters in the vehicle license plate. In [2], the character recognition is done by extracting the plate region of the license plate by using edge detection algorithms and smearing algorithms. Then image segmentation is done by using smearing algorithms, filtering techniques and some morphological algorithms are also used. Lastly, statistical approach is used to identify the characters in the license plate using template matching. In [3], pulse coupled neural network is used for license plate identification. Marques et al [4] proposed an algorithm based on Max/Min post-processing and maximum correlation to locate the license plate for moving and parked vehicle images. This project will focus on implementing an algorithm in [4], which uses hue, saturation, intensity (HSI) color model to determine the threshold statistically for detecting the candidate regions. Irrespective of the color of the license plate, the candidate regions shall include license plate regions and the geometrical properties are used for classification of the license plates. Predetermined license plate alphanumeric characters shall be used to decompose the candidate regions by using 2-D cross correlation and detect the license plate. The effectiveness of the algorithm will be studied followed by conclusions.

Keywords: HSI Color Model, Color Segmentation, Morphological Processing, 2D Cross Correlation.
5. Introduction

The objective of recognizing a specific object in an image is challenging in digital image processing. With the rapid development in the automobile industry and increasing traffic, intelligent transport systems (ITSs) gained significant interest. Intelligent transport system refers to combining information and communication technology to transport infrastructure with the aim of improving traffic, vehicle wear and tear and fuel consumption. The technology based systems in ITSs are basically divided into intelligent infrastructure systems and intelligent vehicle systems [5]. Computer vision and character recognition algorithms for vehicle license plate recognition (VLPR) are vital modules for intelligent infrastructure systems. Vehicle license plate detection (VLPD) is an active research area under ITS. VLPD should operate online in a real-time basis so as satisfy the needs of ITS. VLPD is a challenging task because different vehicle images are taken in different sets of conditions like non-uniform outdoor illumination during data acquisition, different forms of vehicle license plate formats and orientation of the vehicle license plate in the image due to difference in the angle subtended between the camera and object of interest. VLPD is widely used to detect speeding cars, security control in restricted areas and for traffic law enforcement. Detecting the license plate region and performing character recognition is an important research topic of VLPR system.

Generally LPR algorithms are composed of three processing steps that include locating the license plate in the given image, segmentation of the plate characters and recognition of each character. The first two steps are implemented using image processing on still images or videos. Due to exponential increase in the technology, recent developments indicate that it takes less than 50 ms for plate detection and recognition with the ability of processing more than 20 frames per second for videos [5]. Although there are numerous techniques reported in literature for LPR, there is still a need for a common platform where different methods can be evaluated based on
performance, execution time, percentage of false alarm, percentage of detection and recognition rate. Hence it is inappropriate to declare which methods give the highest performance.

The focus of this project is to determine the threshold statistically using HIS color model to binarize the image and then perform character recognition in MATLAB. Around 35 sample images of the car were taken almost in the same lighting condition. From the sample data, the mean, standard deviation and variance of hue, saturation and intensity were calculated for white and red pixels in the image. This method detects license plate having a white background with red colored characters engrossed on the plate, when the angle subtended between the camera and the object of interest is almost perpendicular. This means that the license plate should not be tilted. Furthermore the candidate regions are eliminated using area of the license plate and template matching is done to the characters extracted (recognized) from the license plate to identify the characters.

6. Vehicle License Plate Detecting module

The VLPD module consists of three distinct parts. The first part deals with the conversion of the RGB colored image to HSI image, detection of the candidate region i.e. the license plate. The second part defines procedures for refining the candidate regions using edge detection algorithms, labeling and filtering. Geometrical properties such as area of the license plate can be used to filter the candidate like regions. The last part is the character extraction of the license plate and template matching to identify the characters in the vehicle license plate. Figure 6.1 shows the algorithm of the project implementation.
6.1 Color Segmentation

Input vehicle images that are in RGB color module is converted to HSI color module. The candidate regions are found using the hue, saturation and intensity values of the acquired RGB image.

6.2. RGB Color Model

The RGB color model consists of three basic colors namely red, green and blue. Spectral components of these colors are superimposed to get a resultant color. In a general
sense, the RGB color model describes our perception of color. Three types of receptors in the retina of the human eye have peak sensitivities corresponding to these three primary colors. As seen in the figure 6.2, the RGB color model represents colors within a cubic volume defined by orthogonal Red, Green, and Blue axes. Black is at the origin of the coordinate system (R=G=B=0), white is at the opposite corner of the cube (R=G=B=255). The diagonal connecting the black and white corners (gray dashed line) contains the range of neutral gray levels.

Fig 6.2. RGB Color Model [2].

6.3. HSI Color Model:
Typically HSI colors are not described by the percentage of the composition of the primary colors but rather by their hue, saturation and intensity. The saturation is the pureness of the color, the hue is the color itself and the intensity describes the brightness of the color. It can separate all color components using the hue, saturation and intensity values. As seen in the figure 6.3, the HSI color model represents colors within a double-cone space. The hue is represented as angle $0$, varying from $0^\circ$ to $360^\circ$. Adjusting the hue, color will vary from red at $0^\circ$, through yellow at $60^\circ$, green at $120^\circ$, blue at $240^\circ$ and again back to red at $360^\circ$. Saturation $S$ corresponds to the radius that could vary from 0 to 1. When $S$ is 0, color is gray value of intensity 1. When $S$ is 1, color is on the top of the conical base. On any horizontal slice through the model space, the hue (or “color” of the color) varies around the slice, and the saturation (the purity of the
color) increases radially outward from the central intensity axis. The vertical axis is intensity, which represents variations in the lightness and darkness of a color. The zero intensity level is black while the full intensity level is white. HSI values elsewhere along the intensity axis represent different levels of gray. In the HSI color model, intensity makes no contribution to the color.

Figure 6.3. HSI Color Model [2]

Figure 6.4 shows the RGB car image.

Figure 6.4 RGB Car Image
The following transformations equations are employed to transform an RGB image to HSI image are as below.

\[
H = \cos^{-1}\left\{ \frac{0.5[(R - G) + (R - B)]}{[(R - G)^2 + [(R - B)(G - B)]^{1/2}} \right\}
\]

(6.1)

\[
S = 1 - \frac{3}{(R + G + B)} \min(R,G,B)
\]

(6.6)

\[
I = \frac{R + G + B}{3}
\]

(6.3)

Figure 6.5 shows HSI car image.

![Fig 6.5. HSI Car Image](image)

In the implementation of this project, the license plate detection is based on its color properties namely mean and standard deviation values of hue. To detect white pixels in the license plate, hue has got no meaning. Binarization results in an image whose pixels have only two possible values, 0 (black) and 1 (white) (Fig 6.6). The thresholding method used here is the Otsu algorithm [14][15], which assumes that the image to be
thresholded contains two classes of pixels then calculates the optimum threshold separating those two classes so that their combined spread is minimal. Hence saturation and intensity parameters are taken into account. To determine mean and standard deviation, 35 sample images were taken. Mean, standard deviation of saturation and intensity is computed for white pixels in the sample set of images whereas mean and standard deviation of hue is computed to detect red pixels in the license plate from the same sample of images. For detecting the red pixels, the binarization procedure can be formulated as below,

\[
b_{\text{RED}} = \begin{cases} 1, & \left[ \mu_H - \sigma_H \leq H(x,y) \leq \mu_H + \sigma_H \right] \& \left[ S(x,y) \geq 0.03 \right] \& \left[ 0.05 \leq I(x,y) \leq 0.95 \right] \\ 0, & \text{otherwise} \end{cases}
\]  

(6.4)

where, \( H(x,y), S(x,y) \) and \( I(x,y) \) are hue, saturation and intensity components of the \( x^{th}, y^{th} \) pixel respectively. \( \mu_H, \sigma_H \) are the mean and standard deviation of hue from the data sample respectively.

The binarization procedure for detecting the white license plate pixels is formulated below where \( S(x,y), I(x,y) \) are saturation and intensity components of \( x^{th}, y^{th} \) pixel respectively. \( \mu_S, \sigma_S \) are the mean and standard deviation of saturation values from the data sample respectively while \( \mu_I, \sigma_I \) are the mean and standard deviation of intensity values from the data sample respectively. Figure 6.6 shows binary image after color segmentation.

\[
b_{\text{WHITE}} = \begin{cases} 1, & \left[ S(x,y) \leq \mu_S + \sigma_S \right] \& \left[ I(x,y) \geq (\mu_I + 0.25 + \sigma_I) \right] \\ 0, & \text{otherwise} \end{cases}
\]  

(6.5)

Color segmentation can result in detection of all possible candidate regions from the input image. Some false candidates will be filtered out in the labeling and filtering stage.
6.4. Morphological Processing

After segmentation, there may be some noise in the image such as small holes in the candidate regions. This could be resolved with morphological processing. Mathematical morphology operations are based on the shape in the image and not pixel intensities. The two basic morphological operations available in MATLAB are dilation and erosion [8]. Dilation allows objects to expand while erosion shrinks the objects by eroding the boundaries. These operations can be modified by proper choice of the structuring element which determines how many objects will be dilated or eroded. Structuring element is simply a matrix of 0s and 1s that could be of any arbitrary shape and size. In MATLAB one can define neighborhood of desired size for the structuring element such as square, rectangle, diamond etc. Preferably rectangle is used as the neighborhood for the structuring element of size 6x4 which is obtained by a trial and error method. In the project, closing operation is used which is dilation followed by erosion. Removal of small holes plays an important role in obtaining the rectangular license plate. Figure 6.7 shows morphological operation on binary image.
6.5. Edge Detection, Labeling and Filtering:

Edge detection is one of the important tasks for digital image processing. Edge points possess high gradient difference in the local neighborhood. They are used for feature extraction in image processing. Considerably the image data is reduced after edge detection. Some popular edge detectors [10] are Sobel, Robert’s Cross, Prewitt, Kirsch, Laplace and Marr-Hildreth. Edges are defined as intensity gradients within the image. In this project, Sobel edge detection operator is used in MATLAB. Sobel operator is a combination of two operators: one which can detect horizontal edges and the other which can detect vertical edges. It is a 3x3 neighborhood based gradient operator. The result of the edge image generated by Sobel operator is shown below in figure 6.8.
After detecting the candidate regions using color segmentation and edge detection, features of the region are to be extracted to differentiate LP regions from others. So the next step of the algorithm is to label each of the connected components. Before that the image is smoothened so as to reduce the number of connected components. To smoothen an image is to create an approximate image that attempts to capture important patterns in the data, while leaving out noise or other fine-scale structures/rapid phenomena. The smoothened image is as shown in figure 6.9.
The connected components are labeled. It works on one connected component at a time and can move all over the image. During this step, area of the connected component is used to filter candidate like regions for detecting the license plate. A trial and error method was used to study each connected component namely its area, aspect ratio. This could be used to filter candidate like regions in the license plate so as to obtain the rectangular region as shown in figure 6.10.

![Fig 6.10. Candidate Region](image)

From the above image region of interest is identified as shown in the figure 6.11. The height of the rectangular portion in the image is 31 pixels and the width of the rectangular portion in the image is 71 pixels.

![Fig. 6.11 Region of Interest](image)
6.6. Character Recognition in Vehicle License Plate:

After finding the region of interest from the vehicle license plate, the individual characters can be extracted as shown in fig 6.13. The Region of Interest is divided equally such that each section contains an alphanumerical character. The distance between every character within the alphabets on the license plate is one pixel each. The distance between the third character and the first number on the license plate is ten pixels. Again the distance between every character within numbers on the license plate is one pixel each. After extracting the characters from the vehicle license plate, a template matching technique can be used to identify the characters extracted from the vehicle license plate.

![Character Extraction from License Plate](image)

Fig 6.12. Character Extraction from License Plate

Template of all alphabet and numbers can be collected from the database that is of size 42x24 pixels (Fig 6.14).
The 2-D cross correlation coefficient is calculated between each of the 36 templates with the character recognized in order to identify the character. There are a total of 36 templates which include all the 26 alphabets (A-Z) in English along with numbers (0-9). While computing the 2-D cross correlation coefficient, it should be ensured that image size of the character recognized and the template size should be the same. If they are not of the same size, then there is a need to resize them which can be done using MATLAB. The 2-D cross correlation coefficient is given by,

\[ r = \frac{\sum \sum (A_{mn} - \bar{A})(B_{mn} - \bar{B})}{\sqrt{\left( \sum \sum (A_{mn} - \bar{A})^2 \right) \left( \sum \sum (B_{mn} - \bar{B})^2 \right)}} \]  

(6.5)
where \( r \) is the 2-D cross correlation coefficient, \( A \) is the matrix representing the character identified and \( B \) is the matrix representing the template to be matched. The 2-D cross correlation coefficients is all stored in an array. The maximum value of the 2-D cross correlation coefficient is used to identify the character. The index of the array containing the maximum value of 2-D cross correlation coefficient can be matched with the help of look up table to identify the character.

![Graph](image)

Figure 6.14: shows plot of 2D-Cross Correlation Coefficient of one character recognized with 36 templates.

The above figure 6.13 shows the plot of the 2-D cross correlation coefficient of one character recognized with a total of 36 templates. The peak value of the 2D cross correlation coefficient is identified and the index value on the x-axis gives the alphabet.

7. Experimental Results:

All simulations are performed on Pentium IV processor with Dual core of 3.4 GHz with 2GB RAM under MATLAB. The images were taken using 14 MP camera that are of size
4288 x 3216 pixels. Due to memory requirements of MATLAB, the images are reduced to a smaller size such as 500 x 436 pixels. The images were all taken in the same lighting conditions under bright sun shine. The images were taken at a distance of few meters from the car and the camera focused the license plate region. An accuracy of 96% was achieved in identifying the license plate for a total of 35 images. In [8], the accuracy is reported as 94% for detecting the license plate under different illumination conditions. A common drawback of this method is the failure to detect the boundaries of the license plates. This typically occurs when the license plate and the car are of the same color. This can be negotiated by adjusting the threshold values in the color segmentation process. On an average this method takes 3 seconds to recognize the characters in the license plate. There is an accuracy of 91.6% for identifying the characters of the vehicle license plate for those license plates that have been detected correctly. Sometimes, due to noise in the image of the characters recognized, there can be a possibility of an error in identifying the character.

It is observed that techniques that are based on a combination of edge statistics and mathematical morphology features gave good results [15] for binary image processing. A drawback of this technique is that when edge detection is used to detect the license plate region, it may be too sensitive to unwanted edges due to high local variance that are not of desired interest.

In Hongliang et al [16], a hybrid extraction algorithm based on edge statistics and morphology was developed. This approach consisted of four basic sections: vertical edge detection, edge statistical analysis and morphology based license plate extraction. It was able to achieve an average accuracy of 99.6% for locating a vehicle license plate. The digital images were obtained from a fixed distance, angle and therefore candidate regions in a specific position were given high priority. This prior knowledge can enhance the accuracy to a great extent. Accordingly to [17], if the vertical edges of the car image are extracted with most of the background edges removed, the plate area can easily be
located in the whole edge image. The success rate reported in this paper was found to be 97%. There is still a need for a common platform where different methods can be evaluated based on performance, execution time, percentage of false alarm, percentage of detection and recognition rate.

8. Conclusions:
In this project, a statistical threshold is determined using HSI model of the image in order to obtain the binary image of the car. Geometrical properties such as area, aspect ratio are used to eliminate candidate like regions in the license plate. The algorithm gives good results on our database, and it is relatively robust to variations of the luminance conditions and different kinds of vehicle in different complex environment.

9. Future Work:
There is room for the algorithm to be optimized and it can be further improved by considering more parameters and introducing pre-processing and post processing steps that can be applied to increase the accuracy of detection. The scope of the project can be extended to consider independent orientation.
References


