VEHICLE LICENSE PLATE EXTRACTION AND RECOGNITION

SUDIP ROY
DEPT. OF COMPUTER SCIENCE AND ENGINEERING,
INDIAN INSTITUE OF TECHNOLOGY, KHARAGPUR.
2007

UNDER THE GUIDANCE OF
DR. T. SRIKANTHAN
MR. RAVI KUMAR SATZODA
MRS. SUCHITRA SATHYANARAYANA

PROJECT UNDER CHIPES,
NANYANG TECHNOLOGICAL UNIVERSITY,
SINGAPORE

Acknowledgement

I would like to express my deepest gratitude to **Dr. T. Srikanthan** for giving me this opportunity to work under him in the Centre for High Performance Embedded Systems. I am greatly indebted to him for all his creative ideas and inspirations which give me an impetus to come out with constructive creations.

I would also like to express my gratitude to my immediate supervisors **Mr. Ravi K. Satzoda** and **Mrs. Suchitra Sathyanarayana** for their intellectual guidance, constant attention and unceasing encouragement during the research and development of this project. I benefit much from them.

Thanks to all the staff of CHIPES for their support during my two months stay in Singapore.

Special thanks to my parents and my brother for their consideration, support and encouragement during my stay at NTU, Singapore.

TABLE OF CONTENTS

Chapter 1
Introduction

1.1 Motivation

License Plate Recognition System (LPRS) is a challenging problem in the field of machine vision and automation with various applications including law enforcement, parking lot ticketing systems, automated hands free toll collection, automated vehicle access in secure establishments etc.

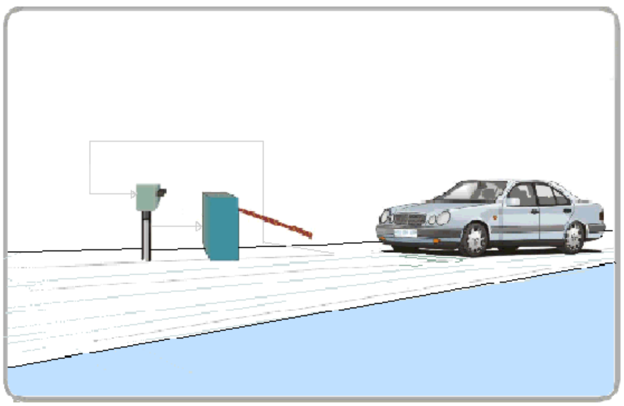


Fig 1.1 Automated Hands free ticketing system (courtesy: The Vision and Image Science Laboratory, Israel Institute of Technology)

A lot of research work has been done in this field, but none of them propose an efficient real time low cost device. Almost all of the proposed work make use of neural networks for the identification purpose and cater to specific dimensions, font, lighting and other conditions. This served as a motivation to develop a low cost smart camera with a dedicated hardware unit for the extraction and identification of the vehicle license plates.

1.2 Objective

The objective of the project was to design a portable hand held camera which can be used to identify license plates. It was intended to simulate the design on software before considering the hardware implementation. This report highlights the software implementation of the design keeping in mind the hardware constraints (both area and time).

In view of this it was intended to use a single engine for both the extraction and recognition purposes. Since the Hough transform has an easy implementation in terms of hardware (using CORDIC arithmetic), and the font used in license plates more or less consists of straight lines, which are highlighted as peaks in the hough domain, an attempt was made to develop the LPRS using hough engine.

1.3 Organization of the Report

This report is organized into 6 chapters with each chapter dedicated to a specific topic.

Chapter 1 presents the motivation, objective and organization of the report.

Chapter 2 presents the literature survey which contains the gist of all the relevant papers related to this topic, various approaches, the pros and cons of each approach and their computational overhead.

Chapter 3 presents the optical character recognition engine using Hough transform. It contains the basic principles involved and the implementation details.

Chapter 4 presents the license plate extraction and segmentation system. It contains the implementation details and a comparison with other works in terms of computational aspects and test set.

Chapter 5 presents the recognition system for the segmented out characters. The various approaches tried out and their pros and cons have been mentioned.

Chapter 6 contains the conclusion justifying the extent to which the objective was realized and the scope of future work in this field.

Chapter 2
Literature Survey

Much work has been done in the fields of optical character recognition and vehicle license plate recognition systems. Below are presented some of the relevant works in both the fields, including the pros, cons and the computational aspects of each of the approaches.

2.1 Hough transform for straight line and curve recognition

Rosenfeld has described an ingenious method due to Hough for replacing the original problem of finding collinear points by a mathematically equivalent problem of finding concurrent lines. This method involves transforming each of the figure points into a straight line in a *parameter space*. The parameter space is defined by the parametric representation used to describe lines in the picture plane [LI2].

Suppose we have *n* figure points and we want to find a set of straight lines that fit them. We transform the points (x_i, y_i) into the sinusoidal curves in the θ-ρ plane defined by

$$\rho = x_i cos\theta + y_i sin\theta.$$

It is easy to show that the curves corresponding to collinear figure points have a common point of intersection. This point in the θ-ρ plane, say (θ_0, ρ_0), defines the line passing through the collinear points. Thus the problem of detecting collinear points can be converted to the problem of finding concurrent curves.

[LI2]Interesting properties of the point-curve transformation:

i) A point in the picture plane corresponds to a sinusoidal curve in the parameter plane.

ii) A point in the parameter plane corresponds to a straight line in the picture plane
iii) Points lying on the same straight line in the picture plane correspond to lines through the same point in the picture plane.
iv) Points lying on the same curve in the parameter plane correspond to lines through the same point in the picture plane.

In principle the transform method extends to arbitrary curves. We need only pick a convenient parameterization for the family of curves of interest and then proceed in the obvious way. A parameterization having bounded parameters is obviously preferable, although this is not essential. It is much more important to have a small number of parameters since the accumulator implementation requires quantization of the entire parameter space and the computation grows exponentially with the number of parameters.

A combination of the application of the novel CORDIC algorithm to the Hough transform and reconfigurable technology has been used to propose a parameterisable Hough transform with real time processing throughput.[LI3]

Various other methods have been proposed for the fast hardware implementation of Hough transform.[LI4][LI7][LI9]
Information regarding the implementation details and variations such as smoothing the accumulators and comparison on the basis of computational complexity between hough transform and template matching can be found in [LI1].

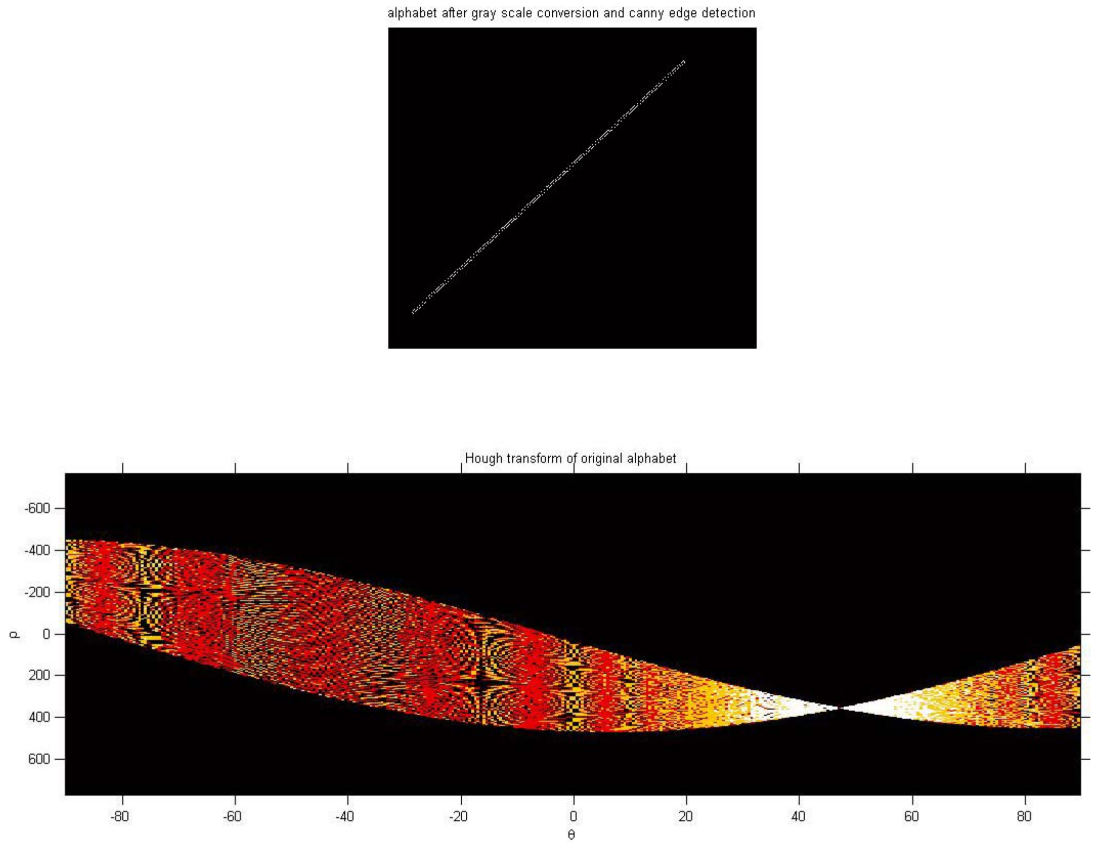


Fig 2.1(a) The edged image of straight line and its 2D hough transform. The line is transformed into a peak which appears as a yellow spot.

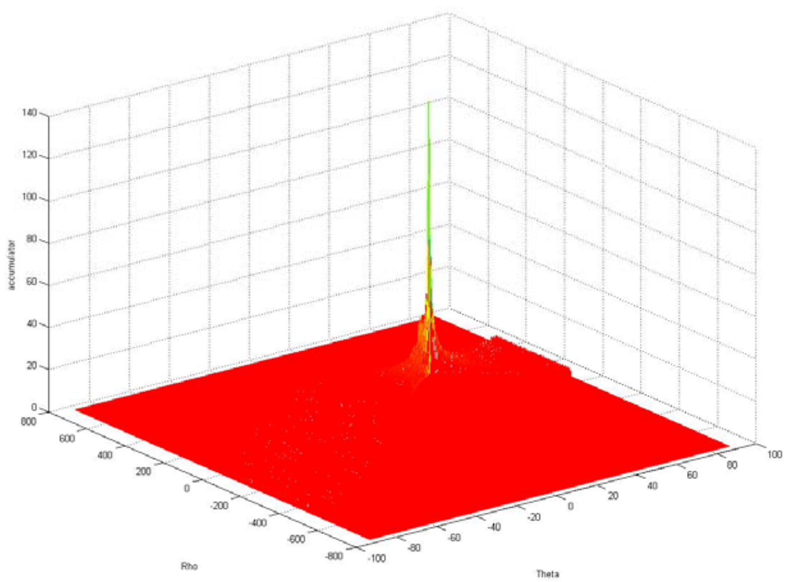


Fig 2.1(b) 3D representation of the Hough. The peak corresponds to the straight line in the picture domain

Circular Hough transform[LI5][LI6] can be used for the identification of circular objects such as coconuts. The disadvantage of this method is that it is highly effective if the (approximate) radii of the objects to be detected are known previously. In case this data is not available CHT has to be implemented for a range of radii which will then make it computationally expensive.

Fig. 6: Successful detection of single coconut

2.2 Hough Transform for Character Recognition

A detailed study of the paper [LII1] throws light on the *compute intensiveness* of the 11 approaches for character recognition. Most require either some intensive preprocessing or some compute intensive logic such as neural networks or fuzzy systems.

Hough transform (both generalized and straight line) has been used for the recognition of Arabic characters.[LII2][LII3]Generalized hough transform requires a multidimensional space which makes it intensive both in terms of memory requirement as well as computation and area[LII2]. The straight line hough which has been used to identify Arabic characters uses hough only to extract features and then do a template matching with features extracted from known characters. This is far less simple than training a neural network as that involves high number of multiplications. No literature could be found on the recognition of English alphabet using Hough or any other hardware friendly way.

2.3 Rotation Invariant Character Recognition

[LIII3]A shape signature, called the scalable translation invariant rotation-to-shifting (STIRS) signature, is obtained from the θ-ρ space by computing the distances between pairs of points having the same θ value. This is equivalent to computing the perpendicular distances between pair of parallel tangents to the curves. The signature has the following properties:1)It is invariant to translation 2) rotation in the image space corresponds to circular shifting of the signature 3) the signature can be easily normalized. Matching two signatures only amounts to computing a 1-D correlation.

Cons: 1) 1-D correlation can be expensive if the array is long
2) It can only be used to match 2 almost exact shapes. So in case of character recognition (where different fonts are involved) there may be many signatures for a particular character.
3) Repetitive patterns in the same image may complicate matters

2.4 Vehicle License Plate Extraction and Recognition

Many models have been proposed for extraction and recognition separately but very few of them propose both an extraction as well as recognition. Also very few papers have paid attention to the hardware computational aspects of the problem. Many of the papers rely on the fact that the number plate edges are properly detected and that the ratio of the width to height of the number plate is a constant.[LIV1] [LIV6] Some of the papers try to apply morphological operations to highlight the potential regions of the number plate and subsequently filter out all but one.[LIV2] Various other methods such as color edge detection and fuzzy systems[LIV5], edge statistics and morphological operations[LIV2], and weight based density maps[LIV9]. In all of the above papers a trade off between accuracy and simplicity (thus time) is observed. For a successful implementation a optimum balance between the 2 factors is highly necessary.

2.5 Concluding Remarks

Thus the survey of the past works emphasizes the need of novel techniques for both recognition and extraction which are simple, hardware implementable and fast. An effort has been made in this report to put forward such a system.

Chapter 3
Optical Character Recognition Using Hough Transform

3.1 Introduction

The past works in the field of character recognition have already been discussed in sec 2.2. From the discussion it is clear that most of the works done in this field are not suitable for hardware implementation. Also no existing literature could be found claiming recognition of English alphabet using Hough transform. Neither could any recognition hardware engine for real time recognition of English alphabet be found. This emphasizes the importance of such a system. An optical character recognition system for the recognition of upper case English alphabet using Hough transform has been proposed in this chapter. The details of the implementations have been discussed as far as possible. The design has been kept as simple as possible such that in future it can be easily implemented in hardware without much difficulty.

3.2 Certain assumptions regarding the input alphabets

The following is a list of certain assumptions:-

1. The algorithm was implemented for a particular font although it seems feasible to make it robust for many other fonts by including minor modifications.
2. Assumed that the input image is a valid uppercase English alphabet
3. Input image is a single pixel width image written in black on a white background

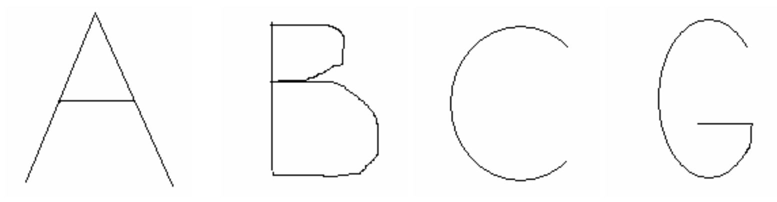


Fig 3.1(a),(b),(c),(d) Examples of valid input images.

3.3 Implementation Details

Out of the 26 uppercase English alphabets 15 namely (A,E,F,H,I,K,L,M,N,T,V,W,X,Y,Z) consist only of straight lines which are oriented to each other at particular angles which might serve as a characteristic signature. The rest of the 11 characters namely (B,C,D,G,J,O,P,Q,R,S,U) consist of curves along with straight lines.

3.3.1 Necessary Preprocessing

A Canny edge detection is done to extract out only the edges as well as to binarize the image. A further crop is done to ensure that the input image now consists of only the bounding box of the character. The image is resized so that absolute thresholds can be set. All of the above pre processing can be removed by modifying the design.

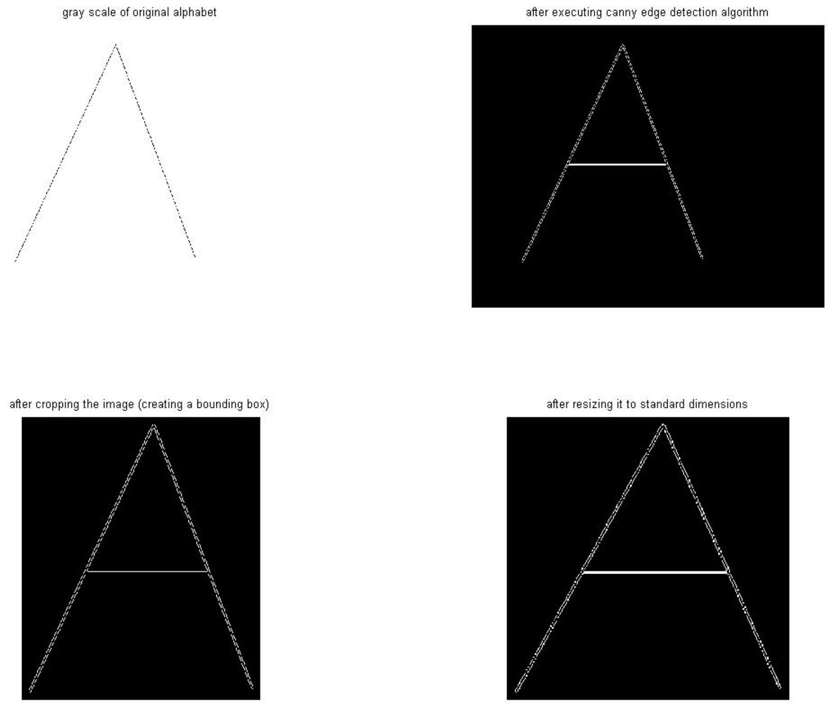


Fig 3.2 (a) Original Gray scale alphabet
(b) Canny edge detected alphabet
(c) After cropping the image
(d) After resizing it to standard dimensions

3.3.2 Curve Identification In Hough Domain – A NOVEL APPROACH

A curve can be mathematically defined to be as a collection of infinite number of straight lines. A curve can be approximated using a finite number of straight lines. This idea is used for the identification of curves. Standard Hough Transforms of curves give a continuous curve of peaks *(*a *crest* or a *ridge)* in the hough domain. We check the values of these peaks at certain discrete values of θ (say -90,-45,0,45 (90 is the same as -90)) and thus approximate every curve to be a collection of four straight lines. The presence or absence and the number of such lines gives enough information to differentiate the curved characters.

3.3.3 Characteristics of Hough Transforms of English Alphabet

A –

1. 3 peaks detected
2. One peak at ±90
3. Other 2 peaks have almost equal ρ and almost equal and opposite values of θ (20 < |θ| <30)
4. |ρ| values for all 3 peaks are almost the same

B -

1. 1 peak value detected at θ = 0
2. Value of ρ very small for this peak
3. Curve Characteristics mentioned later

C –

1. Continuous sinusoidal ridge having –ve ρ values
2. Broken ridge in between -45 to 45 with +ve ρ values

D –

1. 1 peak detected at θ = 0 and ρ almost 0
2. Continuous ridge with +ve ρ values

E –

1. 3 high peaks at θ = ±90 with ρ values having same sign
2. The 3 peaks are equispaced
3. 4^{th} peak at θ = 0 with ρ value extremely small

F –

1. 2 peaks at θ = ±90.
2. One of the peaks has ρ value almost 0
   Other has a value ½ the max value of ρ
3. 3^{rd} peak at θ = 0 with ρ value extremely small

G –

1. 1 peak at $\theta = \pm 90$ with ρ value around ½ the max value of ρ
2. Rest same as those of C

H –

1.2 peaks at $\theta = 0$
One of them has a ρ value almost 0
The other has a positive value of ρ (high)
2. Third peak at $\theta = \pm 90$ with almost half the maximum value

I –

1.3 peaks detected
2. One at $\theta = 0$ with the ρ value non zero
3. 1 peak at $\theta = \pm 90$ with ρ values around the maximum value
4. 1 peak at $\theta = \pm 90$ with ρ values at almost 0

J –

1. 1 peak detected at $\theta = 0$ and high positive value of ρ
2. Two ridges – one spanning -90 to 0 and the other spanning 0 to 90

K –

1. 1 peak detected at $\theta = 0$ and ρ values almost 0
2 2 other peaks at $\theta = \pm 50$
   One with comparatively smaller value of ρ than other

L –

1.1 peak at $\theta = 0$ with ρ almost equal to 0
2. 1 peak at ±90 with very high ρ values

M –

1. 4 distinct peaks
2. 2 of the peaks have ρ values almost equal to 0

3. Arrangement of peaks necessary to differentiate from W

N –
1. 2 distinct peaks at θ around 15 (depends on how N is written)
2. 3^{rd} peak at θ = - 30
3. One of the 2 peaks has a high ρ value

O –
1. 2 parallel ridges (sinusoidal)
Curve characteristics mentioned later

P –
1. 1 peak at θ = 0 with ρ value almost equal to 0
2. One thin ridge of a particular shape

Q –
1. Same as that of O
2. One peak within θ = -60 to -80 (unique to Q only)

R –
1.1 peak at θ almost 0 with ρ value almost 0
2. 1 peak at θ = -45
3. Same ridge as that found in P

S –
Curve characteristics mentioned later

T –
1. One peak at θ = ±90 with ρ almost 0
2. 2^{nd} peak at θ = 0 with mid-ranged ρ value

U –
1. 2 peaks at θ =0 with one of them having ρ almost 0 and the other with a high ρ value

2. 2 ridges one spanning -90 to 0 and the other spanning 0 to 90

V –

1. 2 peaks at opposite sign θ's having difference between 50 to 75
   One of them has ρ value almost o

W –

1. 4 peaks within θ = ± 30
2. Arrangements of peak necessary to differentiate from M

X –

1. 2 peaks at almost equal values of θ with difference of θ values in the range 80 to 100

Y –

1. Similar to that of X (depends on the style of writing)
2. 3 Peaks at 0,-45 and 45

Z –

1. 3 peak points
2. One at – 90 with low ρ value
   One at 45 with mid ranged ρ value
   One at ±90 with high ρ value

3.3.4 Curve characteristics of select alphabet

Angles in degree	±90	-45	0	45
B	3	2	3	2
D	2	1	2	1
P	2	1	2	1
C	2	1 or 2	1	2
O	2	2	2	2
S	3	3	2	1

3.3.5 Classification based on the number of peaks in Hough domain

Class I - 0 peak

C, O, S

Class II - 1 peak

B, D, G, J, P, Q

Class III - 2 peaks

L, V, X, T, R, U

Class IV - 3 peaks

A, F, H, I, K, N, Z, Y

Class V - 4 peaks

M, E, W

The rho values of the peaks at θ=±90 may be required to differentiate between some alphabets.(for eg D and P)

3.3.6 Further classification and identification as implemented in the code

<u>Class I</u>

S – 3 peaks at θ = ±90 (unique to S in this group)

C – Only 1 peak at θ = 0

O – If not C or S then its O (double check possible by checking for 2, 2, 2, 2)

<u>Class II</u>

If there is 1 peak at θ = 0 and the corresponding value of rho is in the low range (< 0.2 * max(rho)) then the alphabet belongs to one of B, D or P
Otherwise it belongs to J, G or Q

B, D, P -

B – 3 peaks at θ = ±90
D – The difference between the rho values of the ridge at θ=±90 is around .5 times the max value of rho
P – If its neither B nor D (additional checks may be imposed)

J, G, Q –

J – 1 peak at θ = 0
G – 1 peak at θ = ±90
Q – If it neither J nor G (additional checks may be imposed)

<u>Class III</u>

Depending on number of peaks at $\theta = 0$

Subclass 1 – no peak

A – 1 peak at $\theta=\pm90$
Z – 2 peaks at $\theta=\pm90$

Subclass 2 – 1 peak

K,Y – no peaks at $\theta=\pm90$
K – the rho value of the peak at $\theta=0$ is small
Y – if its not K (additional checks can be imposed)

F,I – 2 peaks at $\theta=\pm90$

F – the value of rho for the peak at $\theta=0$ is small
I – if its not F (additional checks may be imposed)

Subclass 3 – 2 peaks

N – no peak at $\theta=\pm90$
H – 1 peak at $\theta=\pm90$

<u>Class IV</u>

W,M – no peaks at $\theta=90$

W – the difference between the θ values of the first consecutive peaks is greater than 8

M – if its not W

E – 3 peaks at $\theta = \pm 90$

3.4 Testing the Identification Engine

The identification engine could identify the set of alphabets with minor modifications for which it was designed. To check the robustness of the engine it is necessary to apply it on different fonts. Since the input to the engine is single pixel wide alphabet it is necessary to thin and resize the font alphabet. Thinning code was implemented but the results showed certain distortions in alphabets which the identification engine could not tackle for some cases. Thus either a better thinning algorithm or a modified engine is necessary for exhaustive testing of the design.

The characters which were identified for the font **Futura** are A,C,D,E,F,K,M,N,O, T,U,X,Y,Z . The characters which could not be identified were B,G,H,I,J,L,P,Q,R,S,W.

A Simulink Demo is also presented to show the implementation in an use friendly way.

3.5 Reviewed Classification for a robust engine

Number of peaks in the hough domain

<u>Class I – 0 peaks</u>

C,O,S

<u>Class II – 1 peak</u>

B,D,G,I,J,P,Q

<u>Class III – 2 peaks</u>

B,D,J,L,P,R,T,U,V,X

<u>Class IV – 3 peaks</u>

A,B,D,F,H,I,K,N,P,R,Y,Z

<u>Class V – 4 peaks</u>

B,E,M,R,W

Further classification and identification yet to be done.

3.6 Rotation Invariant Identification Using Hough Transform

A brief discussion regarding this topic has been given in section 2.3 of this report. Please refer to that section for further information. Based on that it can be said Hough Transform has potential to be used for the rotation invariant identification of characters based on the scalable translation invariant rotation-to-shifting (STIRS) signature. It is worthwhile to explore this possibility in the future in projects related to this work.

3.7 Concluding Remarks

It was effectively proved that Hough Transform can be used for the identification of English alphabets. Since the implementation of Hough in hardware is possible in a low cost way, and the identification does not involve any hardware intensive computations, the design is suitable for real time systems.

Chapter 4
Vehicle License Plate Extraction and Segmentation

4.1 Introduction

Vehicle License Plate Extraction involves selection of a frame from a video which possible contains the number plate and then extraction of the region of the number plate. This chapter contains a novel method for the localization of the number plate from a still picture containing the number plate.

A lot of work has been done in this field and a brief gist, pros and cons of each of the approaches have already been discussed in sec 2.4. It can be seen that most of the works focus on software implementation and on only one area that is either plate extraction or recognition, conveniently assuming a successful implementation of the other.

No existing literature is available claiming the use of Hough transform as a principal extraction engine.

4.2 Certain Assumptions over the range of inputs

1. The original image taken is not blurred
2. The image does not have significant background noise such as reflections of trees/vertical panes above the number plate. Such cases have been treated as complex cases and the design can be modified by including some necessary preprocessing to eliminate the effect of such cases in future works.
3. The image should have the number plate of size between 0.1 to 0.5 times the size of the image.
4. The number plate should not be tilted or skewed (-10° to +10°).

5. The background of the number plate is significantly darker (i.e. binarization of the gray levels gives satisfactory results)
6. The characters are written in a font which is clear enough to distinguish the alphabets.
7. The edges in the image are properly detected for all stages.
8. The number plates are single line number plates with the white characters written on black background.
9. The edges of the image are properly detected for all stages.
10. The illumination of the number plate is the same throughout.
11. The input image contains the number plate.
12. The characters are written in white over black background.

Fig 4.1 Pictures of valid test images

Fig 4.2 (a) & (b) have vertical panes, (c) is blurred, tilted and has tree reflections

4.3 Property of the number plate used for recognition – A Novel Idea

Most of the number plates are written in fonts which consist of straight line strokes instead of circular curves. Also there is a sharp contrast between the characters and their background. These 2 properties of the number plates can be used as a distinguishing feature to localize them. A vertical edge detection will give a cluster of vertical line segments (as they are most common in characters) in the region of the number plate. The task is to identify such a cluster taking care not to clip off any part of the number plate.

Such an approach has certain limitations. Some cars have vertical panes just above the number plate which also give a cluster of vertical edges and might be mistaken for number plates. Such cases have been removed from the input image set and can be catered at a later stage.

Only vertical edges are considered as there a lot of horizontal edges in most cars and the number plate will appear as a cluster of only 3 horizontal lines in the Hough domain which will blend it with the background and extraction might not be possible.

4.4 Implementation Details

Given below are the implementation details of the strategy discussed above.

4.4.1 Preprocessing steps

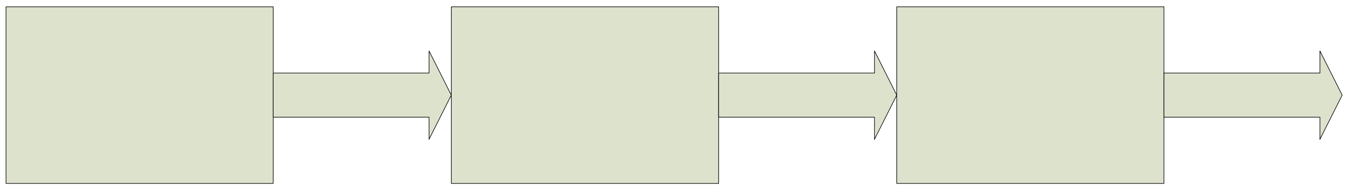


Fig 4.3 Illustration of the preprocessing steps required.

4.4.2 Edge Detection

Sobel Vertical Edge Detection is employed with a heuristic threshold of 30 to detect the vertical edges in the input image. Canny Edge detection is computationally expensive and also includes redundant edges. Prewitt, Zero Cross or any other simple edge detector may also be employed for the purpose.

4.4.3 Slicing Up the Image – A Novel Approach

Initially it has been assumed that the width of the number plate is at least 0.1 times that of the image. Thus we have to detect vertical edges which are at least of that height. Based on these considerations the image is horizontally sliced into 10 strips. Hough transform is applied on each of these strips and the vertical edges (peaks in hough domain within $\theta = -5°$ to $5°$) are searched. The threshold for an accumulator value to be considered as a peak is heuristically set to 12. (this threshold highly depends on the size of the input image) . The number of such peaks is calculated for all the 10 strips. The strip with the highest number of peaks is assumed to contain a major potion of the number plate. The strips to the above and below of this main strip are considered for inclusion depending on the number of peaks detected in those slices with respect to the main slice. 2 strips to the above and 2 strips to the below are considered. Depending on the count of the peaks either a part or the entire strip is included. The ratio of the peaks necessary for a strip to be included and the extent to which it is included have been heuristically set and may not be considered to be the optimum.

All these ratios may be found out from the matlab code files if necessary.

0.6 and above → entire slice include
.33 to .6 → .7 part of the slice included
.2 to .33 → .4 part of the slice included
.2 or less → not included

Since it is necessary to ensure that the entire number plate is included without any clipping an additional padding of 10 pixels is used in some cases when an adjacent slice is not included.

(Another approach could be to identify the slice with the highest peak and consider only the adjacent blocks including above and below in the subsequent steps. This was implemented but was found to be less effective than the above mentioned process)

Fig 4.4 Slice extracted from the image containing the number plate

4.4.4 Extraction of the number plate from the slice using blocks

The slice thus formed is taken and further broken down into 5 blocks vertically. Hough transform is applied on each of these blocks and the number of peaks between $\theta = -5°$ to $5°$ above a particular threshold (set heuristically to 12) is counted. The block with the highest number of peaks is considered to be the primary candidate for the number plate to be present. As in the case of the slices the adjacent blocks are checked for the presence of vertical peaks. A check on the continuity of the vertical edges is also applied by looking at the location of the vertical peaks (i.e . a peak at the ending of one and at another at the beginning of the next suggest a possible continuity and thus the adjacent blocks need to be considered) . All such thresholds are set heuristically and thus might lead to noise inclusion.

Fig 4.5 Primary candidate extracted from a slice

4.4.5 Test Results

1. A Database of 200 images collected from different car parks in NTU was created
2. Existing codes were run on the above mentioned database and the following results were obtained:-

Number of images tested	------------	200
Number of slices correctly detected	------------	146

Failure Analysis:-

Number of images with back-ground noise and reflection noise	-----------	25
Number of images of cars with vertical panes	------------	16
Number of blurred images	----------	7
Number of images rejected on the above basis	----------	48

Total number of relevant test images ----------- 152

Percentage of slice correctly detected ---- 146/152 = 96.05 %

Number of good* crop	-----------	70	→46%
Number of medium** crops	-----------	18	→ 11.8%
Number of poor*** crops	-----------	54	→ 35.5%

*good = entire number plate arrested in crop without much redundancy
**medium = a portion of the first or last character in the number plate is clipped off or the image is over-segmented
***poor =image is under-segmented; not suitable for identification purpose

4.4.6 Inference from the above mentioned results

The method of number plate localization using slices proved successful with a success rate of 96% but the further localization using blocks method proved to be a failure with only 46% of the cases giving satisfactory results acceptable to the recognition engine. Thus it was felt necessary to explore other possibilities to extract out the number plate from the slice containing it. Some of the many things we tried have been enunciated below.

4.4.6.1 Character Length Estimation and direct segmentation

Take the block which is the primary candidate of number plate, perform horizontal trimming by further sub-slicing it, take the horizontal projection to deduce the width of one character, expand either ways in taking blocks of the size of one character and look for characteristic pattern of character, thus find the number plate.

Roadblock: - Difficult to recognize the one character from the horizontal projection because of the possibility of additional noise in the selected block.

4.4.6.2 Number Plate Region estimation based on the black to white pixel ratio continuity

After identification of the primary block use the ratio of white to black pixels as a continuity check to ensure all characters are included in the crop.

Roadblock:- Continuity in the ratio of black to white pixels not decisive in conclusion because of variation in the no of white pixels in the characters i.e. for e.g. 8 has a significantly high number of white pixels.
The number of white and black pixels cannot be taken into consideration as a differentiating parameter as it is difficult to set an absolute threshold to differentiate b/w black and white.

Since the first approach appeared more promising it was decided to stick to it. It was necessary to figure out a way to remove the roadblock mentioned in 4.4.6.1 above.

4.4.7 Sub slicing for noise removal

The number plate thus extracted usually has a lot of noise and thus needs to cropped both vertically and horizontally before it can be made good enough for segmenting out the characters.
The horizontal cropping is achieved by sub slicing. It is the same as the slicing technique applied on the original image again on this smaller image. The image is split into horizontal slices 15 pixels wide and each slice is checked for the presence of vertical edges. The slices are rejected are rejected or accepted on the basis of these number of peaks.
A good horizontal crop is achieved on this basis.

Fig 4.6 Primary Candidate after Subslicing and removing additional noise

4.4.8 Character length estimation and direct segmentation using Horizontal Profile

After the identification of the primary candidate which most probably has a significant portion of the number plate the sub slicing technique is applied to ensure a proper horizontal crop.

Horizontal Profile

It is an array of elements where each element corresponds to one column of the image and its value equals the sum of the pixel values of that column. The horizontal profile show dips because of the inter-character spacing in the number plate. These dips need to be identified. The distance between consecutive dips gives the length of one character. All the characters in the primary candidate can thus be extracted. Some characters such as G,O,D,H,N,…may be split into 2 but can be combined based on the fact that the width of the 2 consecutive sliced up character is too small to be a character. The presence of I I of 11 can be handled later in the recognition stage. The median of the lengths of all the characters was taken and an approximate length of each character was found. Small blocks of the size of the character length so determined were taken on either side of the primary candidate for the number plate till no more characters were found. Thus all the characters in the number plate were extracted along with a few noise blocks.

The thresholds for the peak detection in the horizontal profile have been heuristically set.

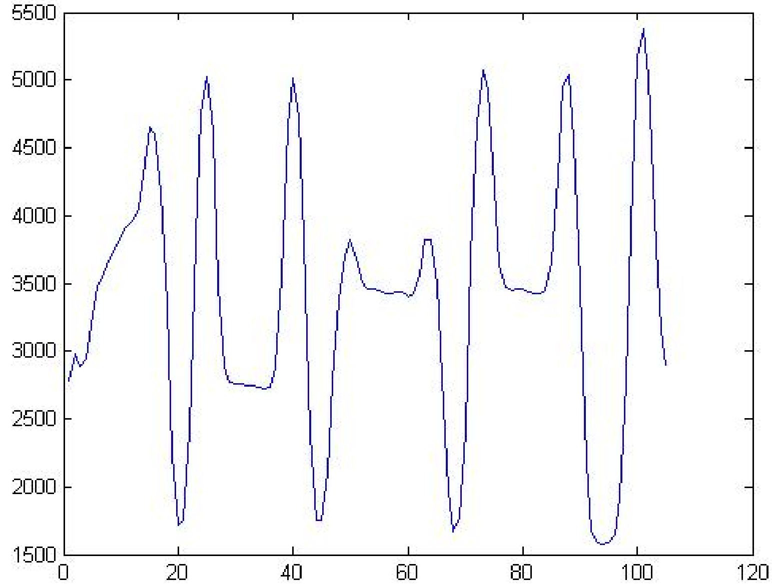


Fig 4.6 (a),(b) Horizontal projection profile of the number plate shown below

4.5 Some of the results

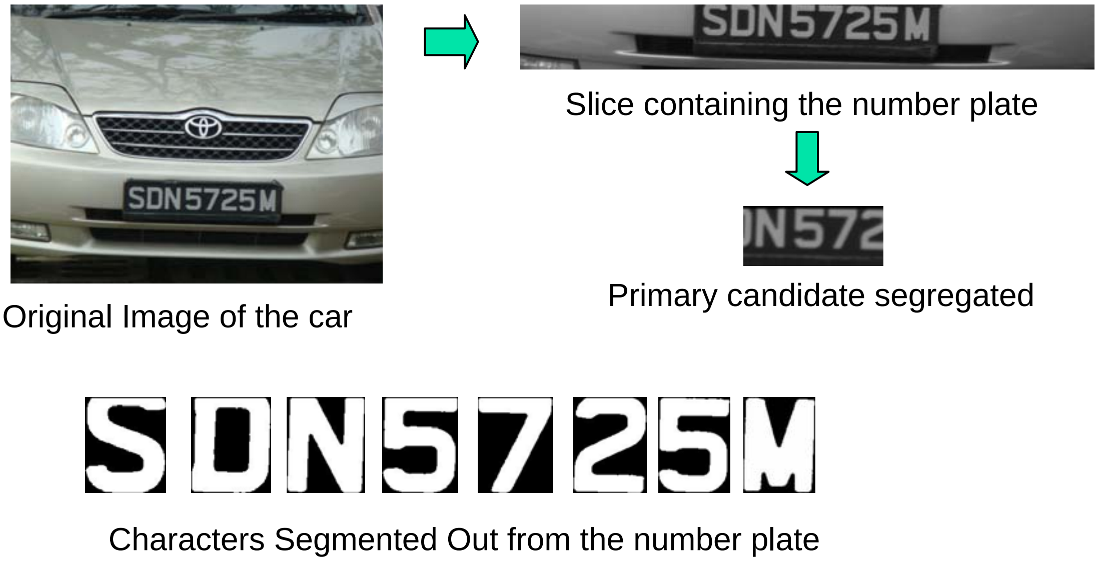
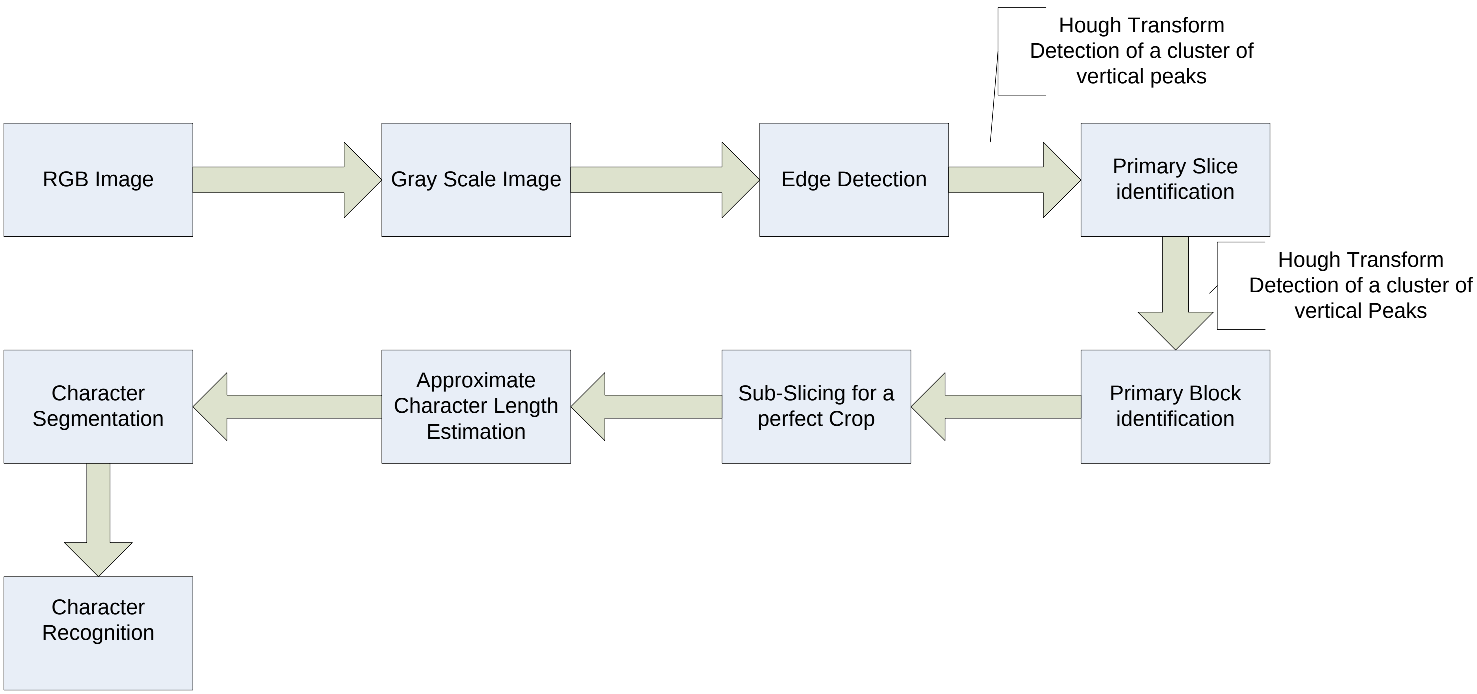


Fig 4.7 The various processing stages

Fig 4.8 Some complex cases illustrating tilt, distance and lighting conditions respectively

4.6 Overall Design Flow



4.7 Concluding Remarks

Thus a novel method of vehicle license plate extraction has been proposed and test results have been shown. The method proposed above uses Hough transform and horizontal projection profile both of which have efficient and fast hardware implementations, to not only extract the number plate but also simultaneously segment out the characters. Thus reducing computational overhead as well as introducing parallelism into the design making it more time efficient.

Further work in this direction is necessary to include all possible complex cases and also consider minor rotation and skew. A robust real time system can thus be developed at low cost.

Appendix

Some of the various things that were tried but proved unsuccessful

1. Conversion of the grayscale image to a 3 scale image. (number plate either appears as white or mid value)
   Code: **gray_2_tri.m**
2. Various contrast enhancement techniques such as contrast stretching, various edge detectors etc. – the problem with most of these was that a universal threshold could not be set.
3. Extraction of the license plate using horizontal profiles on the slices instead of using Hough transforms. – Noise affects to a large extent. Difficult to separate out although visible to human eye
   Code: **contradiction.m**
4. Consider the number of black to white pixels as a continuity measure to identify the number plate. – failed as the black white content of different characters is different.
   Code: **blk_wht_comp.m**

Chapter 5
Vehicle License Plate Identification

5.1 Introduction

The identification part of a License Plate Recognition System has been observed to consume the majority of time. This is because most of the systems implemented so far try to use either neural networks or fuzzy logic systems which are slow and computationally intensive. An effort has been made to propose a fast and simple recognition engine. Though the study is incomplete as substantial results could not be established, yet I think it will be worthwhile to mention the various approaches taken and intermediate results obtained to aid further work in this field.

5.2 The Basic Approach

Since the input to the recognition engine proposed in chapter 3 is a single pixel wide character, it is necessary to convert the segmented out character (which is wide) to single pixel. Most of the thinning algorithms which exist today are computationally intensive and slow as they require multiple number of iterations. An intelligent thinning which suits the purpose of identification of the number plate characters was necessary. (Direct application of Hough transform on the segmented out character gives no information regarding the character as a multiple number of single pixel lines may be drawn. **Effort should be made to develop a modified Hough engine which will identify straight lines of certain width. Such an engine can be easily used for the recognition without any pre processing.)**
Below are enumerated some of the many different approaches taken towards thinning. The pros and cons of the various methods have also been mentioned.

5.2.1 Edge Detection

Applying edge detection on the segmented out characters will give 2 parallel edges. It is difficult to differentiate multiple number of parallel edges (letters such as 8,0,E,F..). Thus the edged images are not suitable for identification. Also the logic involved in recognition will bloat up to twice.

Fig 5.1 Edge (Sobel) of the segmented out characters.

5.2.2 First White Pixel

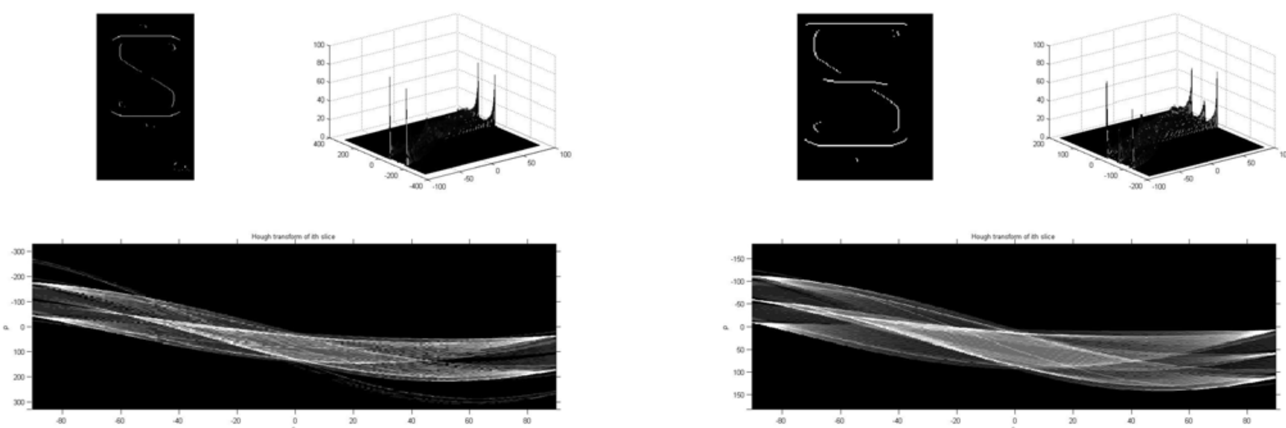
An intermediate representation for all the alphabets can be found by taking only the first white pixel when scanning from bottom to top and from right to left (these 2 directions are chosen because they preserve the shape of the character to the largest extent). The representation gives well defined peaks in the Hough domain. The problem with this method is that there are certain characters which give the same intermediate representation. Some such characters are (O,D,Q,G),(5,6),(B,8),….Computationally this method is quite simple as it does not involve any storage or calculations. Subsequently it was decided to consider all the black to white transition edges. Thus the internal edges of the alphabets were also included adding to the identity of the characters but still the intermediate representation was not unique.

Fig 5.2 The black to white transition pixels when scanning from bottom to top and from right to left

5.2.3 Image Thinning by taking the mid point of the thick edges

A new approach was taken with an aim to retain as far as possible the basic shape and differentiating characteristics of the characters. Instead of taking the black to white transition pixels, the midpoint of the black to white and white to black transition pixel was taken. This although with a slight computational overhead than the previous result did increase the level of uniqueness of the representations. In spite of this, the representation could not cater to the variations in the fonts with which the plates are written and unique feature sets for characters could not be established.

For eg. below shown are various representations of the character S and their corresponding Hough transforms in 2D and 3D.



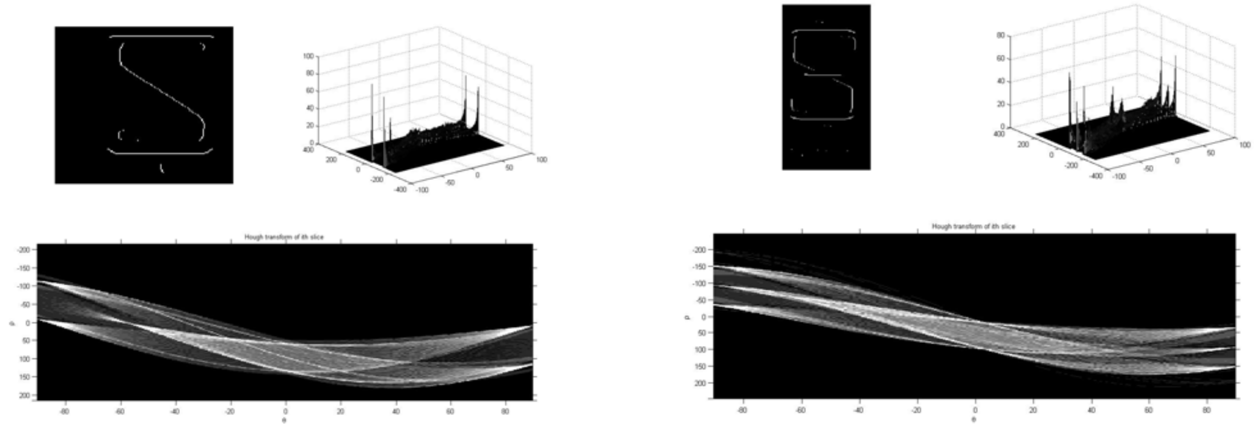


Fig 5.3 Different representations for S making identification difficult

5.3 Bounding Box Extraction

As can be seen in the representations of the characters above the image has not been properly segmented. Thus in order to ensure a proper cut the vertical and horizontal profile of the binary image was taken and the largest continuous stretch of white pixels was identified in both directions. Thus making use of the fact that the characters are connected component and the major constituent of the image an appropriate cut could be achieved.

Fig 5.4 (a),(b),(c) shows the original image, vertically cropped image and both vertically and horizontally cropped image clockwise from top

5.4 Horizontal and vertical Profile

Attempt was made to combine the information from the horizontal profile, vertical profile and the Hough transform to conclusively identify the characters. The study could not be completed due to the lack of time. Further investigation in this direction is necessary and advised. The details of the peak detection algorithms are mentioned in the next section. It might be possible to use the spatial arrangements of the peaks and the relative height of the peaks and the angle of the transversal lines derived from the Hough transform domain to conclusively identify the characters. Various degrees of classification could be achieved depending on which method is used first but no conclusive statement can be made regarding the exact sequence which might lead to an optimum result.

5.5 Peak Detection Algorithms

Calculate the profile (horizontal/vertical) of the clipped character. Each element of the array (profile) has the count of the number of white pixels in that column/row.
Smoothen the curve using a 1x3 averaging mask of [1 1 1] (take the average of the neighbors and itself).
We have to scan through this array to detect the peaks. A preliminary scan is done first to find out all candidate peaks out of which relevant ones are filtered out at a later stage.

5.5.1 Preliminary Scan

Condition for a point to be classified as peak:-

1. The value of the profile of that index is greater than the value of its neighbors.
2. The previous value in the peaks array (array containing the index and value of all the peaks and dips identified till that point) is either a dip or a smaller peak. If the previous one is a smaller peak remove that and replace it with this peak.
   If the previous value is a dip and the difference between it and this is greater than 10 then add this index to the peaks array and mark it as a peak.

Similarly the dips can also be identified and added to the peaks array.
After the preliminary scan the array contains all the index which are possible peaks or dips. All the elements in this array have been marked alternately as peaks or dips.

5.5.2 Relevancy Check

A peak is considered to be relevant if the difference between this and the previous relevant extrema and the next preliminary extrema is greater than threshold value.
Similarly the relevant peaks can also be separated out from the preliminary peaks.
These set of relevant peaks provide vital information (position and magnitude of the peaks) necessary for the identification of the character.

5.6 Concluding Remarks

A combination of the above methods and a recognition engine similar to the one proposed in chapter 3 can be used to effectively complete the entire system. A complete software simulation will be a tremendous boost towards hardware implementation and realization.

Chapter 6
Conclusions and Future Work

6.1 Conclusions

The work was aimed at proposing efficient, fast, simple and hardware friendly vehicle license plate recognition system. The 2 modules of the project namely i) license plate localization and segmentation ii) character recognition have been individually realized. Both the modules use the same basic principle of Hough transform and thus can be implemented in a pipelined manner, using the same Hough engine. This would ensure a cheap hardware. The simplicity of the design ensures a fast and real time system. Apart from the ones found successful various other methods to achieve the same have been studied and implemented. This report forms the foundation stone of further work.

6.2 Future Work

6.2.1 License Plate localization

The system proposed in this report does not analyze or cater to complex cases such as varying degree of illumination, occlusion, rotation, skew, multiple number plates, random noise manifesting itself as a cluster of vertical straight lines etc. All such cases need to be studied and evaluated. Also to propose a real time system frame grabbing from a continuous video and choosing the frame containing the number plate is necessary. Work in this direction could also be possible done in the future. An exhaustive set of test image database is necessary to confidently put forward the results. Some of the number plates might get clipped. Although rare this problem needs to be tackled. Various other possible methods for the license plate extraction may also be explored.

6.2.2 License Plate Identification

Out of the various methods for identification proposed in chapter 5, an appropriate combination must be put forward to ensure correct identification with least computational overhead. Also the possibility of identification from the binarized image directly without any pre processing by using a modified hough engine which identifies only wide lines instead of identifying all single pixel lines must be explored thoroughly.(something which I personally believe is possible and worth exploring).

After the extraction and segmentation is made robust and an identification engine is developed in software, the next step would be to do a hardware complexity analysis and then move on to implement the design so as to design a dedicated chip which can be appended to cameras making them smart for number plate identification.

This work could be merged with other works involving intelligent transport systems and machine vision such as that of lane detection using hough transform and may be used to recognize warning signs written on the road or sign boards provided on the paths.

Best wishes to the one who continues this work. Please feel free to contact me for further information or clarification.

Thanking You
SUDIP ROY

BIBLIOGRAPHY

[LI1] Bryan S. Morse, "*Lecture on segmentation using hough transform*", 2003

[LI2] Richard O. Duda and Peter E. Hart, "*Communications of the ACM*", January 1972, Volume 15, November 1

[LI3] Dixon D. S. Deng, Hossam ElGindy, "*High-speed parameterisable Hough Transform Using Reconfigurable Hardware*", Pan-Sydney Area Workshop on Visual Information Processing (VIP2001), Sydney, Australia.

[LI4] Sandy Pavel, Selim G. Akl, "*Efficient Algorithms for the Hough tranform on Arrays with Reconfigurable Optical Buses*", The research was supported by the Telecommunications Research Institute of Ontario and Natural Sciences and Engineering Research Council of Canada..

[LI5] Mohamed Rizon, Haniza Yazid, Puteh Saad, Ali Yeon Md Shakaff, Abdul Rahman Saad, Masanori Sugisaka, Sazali Yaacob, M. Rozalin Mamat and M. Karthigayan, "*Object Detection Using Circular Hough Transform*", American Journal of Applied Sciences 2(12):1606-1609,2005

[LI6] Jau Ruen Jen, Mon Chau Shie, and Charlie Chen, "*A Circular Hough Transform hardware for Industrial Circle Detection Applications*", ICIEA 2006 (IEEE)

[LI7] Robert Strzodka, Ivo Ihrke, Marcus Magnor, "*A Graphics Hardware Implementation of the Generalized Hough Transform for fast Object Recognition ,Scale, and 3D Pose Detection*"

[LI8] Cvetan Cheshkov, "*Fast Hough Transform Algorithm*", HLT Workshop 2004,CERN

[LI9] Samir Tagzout, karim Achour and Oualid Djekoune, "*Hough Transform algorithm for FPGA Implementation*", IEEE 2000

[LII1] Oivind Due Trier, Anil K. Jain, and Torfinn Taxt, "*Feature Extraction Methods for Character Recognition – A Survey*", Pattern Recognition, Vol. 29. No. 4, pp. 641-662, 1996

[LII2] Sofien Touj, najou Essoukri Ben Amara, Hamid Amiri, "*Generalized Hough Transform for Optical Character Recognition*" , Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR 2003)

[LII3] Mohamed Fakir, M. M. Hassani, Chuichi Sodeyama*, " On The Recognition of Arabic Characters Using Hough Transform Technique"*, Malaysian Journal of Computer Science, Vol. 13, No. 2, December 2000, pp.39 – 47

[LII4] P.M.Patil, P.S.Dhabe, U.V.Kulkarni and T.R.Sontakke, *"Recognition of Handwritten Characters using Modified Fuzzy Hyperline Segment Neural Network"*, The IEEE International Conference on Fuzzy Systems,2003

[LIII1] Derek C. W. Pao, Hon F. Li, R. Jayakumar, *"Shapes Recognition Using the Straight Line Hough Transform: Theory and Generalization"*, IEEE Transactions on Pattern analysis and Machine Intelligence, Vol 14, No. 11, Novemeber 1992.

[LIII2] Xiaofan Liu, Shaohua Tan, V. Srinivasan, S. H. Ong, *"Selection of Key Features for Invariant Object Recognition Using Fuzzy Entropy"*, 1994 international Symposium on Speech, Image Processing and Neural Networks,13-16 April 1994,Hong Kong.

[LIII3] G.Y.Chen, T.D.Bui, A. Krzyzak, *"Rotation invariant feature extraction using Ridgelet and Fourier transforms"*, Pattern anal Applic (2006) 9: 83-93

[LIII4] Michael Villamizar, Alberto Sanfeliu and Juan Andrade-Cetto, *"Computation of Rotation Local Invariant features using the Integral inage for Real Time Object Detection"*, The 18th International Conference on Pattern Recognition (ICPR'06).

[LIV1] M.J.Ahmed, M.Sarfraz, A.Zidouri, and W. G. Al-Khatib, *"License plate Recognition System"*, ICECS-2003,(IEEE)

[LIV2] Jun-Wei Hseih, Shih-Hao Yu, and Yung-Sheng Chen, *"Morphology-based license Plate Detection from Complex Scenes"*, IEEE 2002

[LIV3] Choudhary A. Rahman, Wael Badawy , *"A Real Time Vehicle's License Plate Recognition System"*, Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS'03)

[LIV4] Vladimir Shapiro, Georgi Gluchev, *"Multinational License Plate Recognition System: Segmentation and Classification"*, Proceedings of the 17th International Conference on Pattern Recognition (ICPR'04)

[LIV5] Shyang-Lih Chang, Li-Shien Chen, Yun-Chung Chung, and Sei-Wan Chen, *"Automatic License Plate Recognition"*, IEEE Transactions on Intelligent Systems, Vol. 5, No. 1, March 2004

[LIV6] Bai Hongliang and Liu Changping, *"A Hybrid License Plate Extraction Method Based On Edge Statistics and Morphology"*, Proceedings of the 27th International Conference on Pattern Recognition (ICPR, 04)

[LIV7] Dai yan, Ma hongqing, Liu Jilin and Li Langang, *"A High Performance License Plate Recognition System Based On The Web Technique"*, 2001 IEEE Intelligent Transportation Systems Conference Proceedings

[LIV8] Nikolas Bellas, Sek m. Chai, Malcom Dwyer, Dan Linzmeier, *"FPGA implementation of a license plate recognition SoC using automatically generated streaming accelerators"*, IEEE,2006 (patent pending)

[LIV9] R. Bremananth, Dr. A. Chitra, *" A Robust video Based License Plate Recognition System"*, Proceedings of ICISIP,2005