CORE Metadata, citation and similar papers at core.ac.uk

İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

EMBEDDED DSP BASED LICENSE PLATE LOCALIZATION

M.Sc. Thesis by

Özgür BULKAN, B.Sc.

Department :	**Computer Engineering**
Programme :	**Computer Engineering**

JUNE 2008

İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

EMBEDDED DSP BASED LICENSE PLATE LOCALIZATION

M.Sc. Thesis by
Özgür BULKAN, B.Sc.
(504001577)

Date of submission : 5 May 2008

Date of defence examination: 10 June 2008

Supervisor (Chairman): Prof. Dr. Muhittin GÖKMEN (İTÜ)

Members of the Examining Committee Prof. Dr. Coşkun SÖNMEZ (YTÜ)

Assoc. Prof. Mustafa KAMAŞAK (İTÜ)

JUNE 2008

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

DSP TABANLI GÖMÜLÜ PLAKA YER SAPTAMA

YÜKSEK LİSANS TEZİ
Müh. Özgür BULKAN
(504001577)

Tezin Enstitüye Verildiği Tarih : 5 Mayıs 2008
Tezin Savunulduğu Tarih : 10 Haziran 2008

Tez Danışmanı :	**Prof.Dr. Muhittin GÖKMEN (İTÜ)**
Diğer Jüri Üyeleri	**Prof.Dr. Coşkun SÖNMEZ (YTÜ)**
	Yrd. Doç. Dr. Mustafa KAMAŞAK (İTÜ)

HAZİRAN 2008

ACKNOWLEDGEMENT

This thesis which is entitled as “Embedded DSP Based License Plate Localization” was performed at the İTU Computer Engineering department. First of all, I would like to thank my supervisor Prof. Dr. Muhittin GÖKMEN for his support, reviewing this thesis and for many fruitful discussions on the results of my work. I would like to thank M.Sc. Muhammet S. BERATOĞLU, research assistant, for valuable contributions to this thesis.

I wish to thank Ph.D. Abdulkerim ÇAPAR and B.Sc. Ramazan ÇETIN for their constructive help with many problems.

Especially, I am more than grateful to my wife, Sevinç KARACA BULKAN, always next to me to encourage and give incredible support along my studies and all my family for their support during all my lifetime.

May 2008 **Özgür BULKAN**

TABLE OF CONTENTS

ACKNOWLEDGEMENT .... **iii**
TABLE OF CONTENTS .... **iv**
ABBREVIATIONS .... **vi**
LIST OF FIGURES .... **vii**
LIST OF TABLES .... **viii**
LIST OF SYMBOLS .... **ix**
SUMMARY .... **x**
ÖZET .... **xi**

1. INTRODUCTION **1**

1.1 Description of the problem **1**

1.2 Basics of License Plate Recognition System **2**

1.2.1 Image Acquisition 2
1.2.2 Memory and Data movement 2
1.2.3 Preprocessing 3
1.2.4 Plate Localization 3

1.3 Related Works **4**

1.4 Proposed System Architecture **5**

1.4.1 Hardware Description 7
1.4.1.1 Video Acquisition 8
1.4.1.2 Video Data Movement 10
1.4.1.3 Internal and External Memories 12
1.4.1.4 License Plate Localization 12
1.4.1.5 System Output 14
1.4.2 Software Environment 14
1.4.3 Advantages of the System Proposed 15

2. LICENSE PLATE LOCALIZATION **16**

2.1 Reference System **16**

2.2 Overview **16**

2.3 Segmentation **17**

2.3.1 Canny Edge Detection 17
2.3.1.1 Good Detection 17
2.3.1.2 Good Localization 18

2.3.1.3 One response to one edge 18
2.3.2 Gradient Image 18
2.3.2.1 Calculation of Gradients 19
2.3.2.2 Discarding adjacent non-extreme values 21
2.3.3 Threshold 22
2.3.3.1 Detecting Edge Pairs through a raster scan 23
2.3.3.2 Threshold with Edge Pairs 25
2.3.3.3 Connected Component Analysis 26

2.4 Detecting Plate Character Candidates 29

2.5 Detecting Plate Candidates 32

3. RESULTS 34

3.1 Sample Results 34

3.2 Experimental Results 39

4. CONCLUSION 45

REFERENCES 46
CIRCULUM VITAE 49

ABBREVIATIONS

LP	: License Plate
LPR	: License Plate Recognition
CPU	: Central Processing Unit
DSP	: Digital Signal Processor
PAL	: Phase Alternating Line
DMA	: Direct Memory Access
PC	: Personal Computer
I/O	: Input and Output
FPGA	: Field Programmable Gate Array
ITU	: International Telecommunication Union
İTU	: İstanbul Technical University
DIVIT	: Digital Video and Image Technologies
SAV	: Start of Active Video
EAV	: End of Active Video
PPI	: Parallel Port Interface
L1	: Internal limited size memory of Harward Architecture
L2	: Internal memory of Harward Architecture
L3	: External memory of Harward Architecture
ALU	: Arithmetic Logic Unit
DAG	: Data Address Generators
MAC	: Multiply Accumulators
JTAG	: Joint Test Action Group
USB	: Universal Serial Bus
ADSP	: Analog Devices DSP Processor
BF561	: ADSP, Model Blackfin, Type 561
RLC	: Run Length Code
RLE	: Run Length Encoding

LIST OF FIGURES

Figure 1.1 : A License Plate Recognition Application on a PC Based System........ 5
Figure 1.2 : DSP Based LPR System - 1.. 6
Figure 1.3 : DSP Based LPR System - 2.. 7
Figure 1.4 : Hardware Design Block Diagram 7
Figure 1.5 : Internal DSP Architecture 8
Figure 1.6 : ITU BT.601/656 Video Data Structure 9
Figure 1.7 : ITU BT.601/656 Serial Stream.. 10
Figure 1.8 : 2D to 1D DMA Access.. 11
Figure 1.9 : ALU, DAU, and Control Unit 12
Figure 1.10: BF561 EzKit.. 14
Figure 2.1 : Convolution Efficiency Sample 20
Figure 2.2 : Calculation of Gradients... 22
Figure 2.3 : Negative-Positive Magnitude Gradient Pairs Analysis 24
Figure 2.4 : Threashod with Negative and Positive Gradient Pairs 26
Figure 2.5 : Blob Coloring Flow Chart... 27
Figure 2.6 : Connected Components of Input Image 28
Figure 2.7 : Surrounding Rectangles of Blobs 28
Figure 2.8 : Plate Character Candidate Detection Flow Chart............................ 30
Figure 2.9 : Output of rectangles after merging and applied restriction rules 31
Figure 2.10: A sample Input Image 31
Figure 2.11: Connected Component Stage 31
Figure 2.12: Plate Candidates Detection Flow Chart.. 33
Figure 3.1 : Interlaced D1 Input Frame.. 34
Figure 3.2 : Horizontal Gradients 35
Figure 3.3 : Segmented Image 35
Figure 3.4 : Reduced Rectangles... 35
Figure 3.5 : Plate Localization.. 36
Figure 3.6 : Results focused on plate on plate 37
Figure 3.7 : Results subject to surrounding region of plate area.......................... 38
Figure 3.8 : Sample Results captured from video stream 40
Figure 3.9 : Results with changing local threshold level 41
Figure 3.10: Sample results versus character stroke length.................................. 42
Figure 3.11: Sample Results versus minimum and maximum character height...... 42

LIST OF TABLES

Table 1.1 Bit weighting for 1.15 Fractional Data Types.. 13
Table 1.2 Some of 1.15 fractional numbers and their decimal equvalents............. 13
Table 2.1 Floating Number of Differentiation of Gaussian Coefficients................ 19
Table 2.2 Fractional Data of Differentiation of Gaussian Coefficients................... 19
Table 2.3 Algorithm Optimizations with ALU operations...................................... 21
Table 3.1 License Plate Localization Results.. 39
Table 3.2 Power Consumption Figure... 43
Table 3.3 Memory Utilization.. 43
Table 3.4 Sample Functions Optimization Results 44

LIST OF SYMBOLS

G	: Gaussian Function
G′	: First Derivative of Gaussian
LOC	: Localization Criterion
T	: Single Response Criterion
h	: First derivative of Gaussian Kernel
Ri	: Surrounding Rectangles of Blobs
L	: Left Coordinate of Rectangle
R	: Right Coordinate of Rectangle
T	: Top Coordinate of Rectangle
B	: Bottom Coordinate of Rectangle
W	: Width of Rectangle
G	: Maximum Distance between rectangles
Y1	: Lower Bound of Height Criterion
Y2	: Upper Bound of Height Criterion
H	: Rectangle Height Criterion

EMBEDDED DSP BASED LICENSE PLATE LOCALIZATION

SUMMARY

The system presented and designed in this work as an embedded DSP architecture corresponding to real time video processing constraints is an application of license plate localization which is a challenging issue and distinctive unit of full featured and considerably standardized automated recognition systems required in several application areas like traffic management, custom controls, toll-pay systems, identification of stolen cars, parking, controlling of restricted zones. The reason of the fact it is a distinctive part of overall recognition system is that the issue is basically reduced to a recognition stage once the location of the license plate is correctly found. Beyond the reason that it is an issue to enhance the performance gain as a very important milestone prior to recognition modules, it is a priority task as a part of typical video surveillance system that the application should propose compact design, portability, low power consumption and low cost architecture as compared with generic personal computer based systems. It is aimed to consider all these constraints in algorithm and system design and development.

Localization Algorithm generally consists of edge detection, threshold, component labeling, determination of surrounding rectangles of plate characters candidates, and finally localization of the plate in an input image. Edge detection problem is reduced to find only horizontal edges sufficient prior to local threshold operation required for reasonable segmentation of interested parts of input image. Connecting component labeling of segmented parts gives surrounding area of objects including plate characters in image. Definition of several basic rules to correctly determine plate characters as eliminating fake objects and localize plate region gives sufficient results for a predefined and feature analyzed database as in literature.

As contemplated, a DSP based embedded real-time video surveillance system is designed and developed comparatively sufficient to generic computer based systems in resolutions of both performance and efficiency constraints in addition to license plate localization algorithm development by utilizing flexible, powerful, complex multifunction instructions, high performance direct memory access and general purpose input outputs and multi core structures of integrated DSP.

DSP TABANLI GÖMÜLÜ PLAKA YER SAPTAMA

ÖZET

Bu çalışmada sözü edilen ve tasarlanan sayısal sinyal işleme tabanlı gömülü bir mimari üzerinde gerçek zamanlı görüntü işleme kıstaslarına uyularak oluşturulan plaka yer saptama uygulaması; trafik yönetimi, gümrük kontrolleri, otoyol ödeme sistemleri, çalıntı arabaların tanınması, park yerleri, yasak bölgelerin kontrolü gibi birçok uygulama alanında gerek duyulan tam işlevli ve özdevimli tanıma sistemlerinin ayırt edici bir özelliği ve ilgi duyulan bir konudur. Tanıma sisteminin bütününün ayırt edici bir parçası olmasının nedeni bir kez plaka yeri doğru olarak saptandığında aslında sorunun tanıma aşamasına indirgenmesidir. Tanıma biriminin girişinde yer alan çok önemli bir aşama olarak başarımındaki kazancı iyileştirme sorunu olması gerçeğinin ötesinde, bilinen bilgisayar tabanlı sistemlerle karşılaştırıldığında uygulamasının işlevine göre dar hacimli, kolay taşınabilir, düşük enerji tüketimli, ve düşük maliyetli bir mimariye sahip olan tipik bir görüntü gözetim sisteminin bir parçası olması öncelikli bir iştir. İşlemsel süreçlerde, sistem tasarım ve geliştirme aşamalarında tüm bu kısıtlamaların göz önünde tutulması amaçlanmıştır.

Yer saptama işlemsel süreci genel olarak ayrıt bulma, eşikleme, bağlı bileşen etiketleme, plaka karakterlerini saran dikdörtgenlerin bulunması, ve son olarak da giriş görüntüsündeki plaka yerinin belirlenmesinden oluşur. Ayrıt tanıma problemi giriş görüntüsündeki ilgilenilen alanların anlamlı parçalara ayrılması için gerekli olan lokal eşikleme işlemi öncesinde yalnızca yatay yönlü ayrıtların bulunacağı bir şekle indirgenmiştir. Parçalanmış alanların bağlı bileşenlerinin etiketlemesi resim üzerindeki plaka karakterlerini de içeren dikdörtgen alanları belirler. Yanıltıcı nesnelerin elenerek, plaka karakterlerinin ve plaka bölgesinin doğru olarak bulunabilmesi amacıyla temel bir takım kuralların belirlenmesi, önceden tanımlanmış ve özellikleri çözümlenmiş bir veri tabanı için yeterli sonuçlar vermektedir.

Öngörüldüğü şekliyle, tümleşik devrenin kullanışlı, yetkin, karmaşık ve çok işlevli paralel çalışan komutları ile yüksek başarımlı doğrudan bellek erişimi, genel amaçlı giriş, çıkış ve çok çekirdekli yapısından faydalanılarak plaka yer saptama işlemsel sürecinin geliştirilmesine ek olarak, sayısal işaret işleme tabanlı gömülü gerçek zamanlı bir görüntü izleme sistemi tipik bilgisayar tabanlı sistemlerle kıyaslandığında hem başarım hem verimlilik açısından gereksinimleri oldukça karşılayacak şekilde tasarlanmış ve geliştirilmiştir.

1. INTRODUCTION

In this thesis, the task of license plate (LP) localization in a License Plate Recognition (LPR) system is presented, designed to work and implemented on an embedded DSP platform to process real time video streams.

All LPR algorithms consist of two major parts as detection and recognition modules which are generally a combination of hardware and software systems. The hardware components are usually generic computers interfaced to video peripheral devices. Algorithms used for both detection and character recognition requires high amount of CPU power that is capability of supporting only a restricted number of video surveillance subsystems.

Today, wide range applications of video surveillance systems including traffic management, gate, custom controls or similar access control of restricted areas make the algorithm and system design of LPR challenging issue for new and enhanced approaches.

In this work, an embedded real time DSP platform is proposed to be designed in addition to effective algorithm development for License Plate Localization process instead of generic personal computer and peripheral based system designs. It aims flexible, powerful, complex multifunction instructions and multi core structures used for optimized algorithm development in an integrated embedded system.

Following parts in this section will firstly give the description and basics of LPR systems, overview of some application areas where its technologies are deployed, related works, and, lastly, functional software and hardware block diagrams of proposed system.

1.1 Description of the problem

A car plate is unique for any vehicle getting authorization in traffic. License plate localization is used to locate license plates from images including a vehicle accurately and efficiently. An automated system to identify vehicles in different scenarios is mostly required in several application areas like traffic management, custom controls, toll-pay systems, identification of stolen cars, parking, control of restricted zones… etc. Automatic identification of a license plate (LP) consists of mainly three steps; isolation of license plate region, segmentation of license plate character regions, recognizing the unique identity string.

As identifying license plates (LP) becomes very important more and more by reason of increasing many security requirements, chaotic road traffics in large cities gets this task intractable in opposite. Today, there is an increased interest to enhance contemporary automated systems well known that the performance is restricted with several conditions like changing illumination, high speed of the cars, noisy pictures, background noise, natural scenes in frame, hidden text in picture, real-time requirements and cost. All makes solution of the task difficult and the system does not work robustly in many situations.

1.2 Basics of License Plate Recognition System

In this section, the general components of a License Plate Recognition (LPR) system are overviewed before proposed system architecture is given.

1.2.1 Image Acquisition

Generally, any image processing systems receives image signals from a camera unit. Likewise, the camera outputs the images or image sequences to the system of LPR. These are in general stand-alone working devices which output stream in a number of different standards correspond to existing video backend devices. In real time video applications, for example, active image resolutions are typically either 720x486 (525/60 video systems) or 720x576 (625/50 video systems) as given PAL ITU-R BT.601/656 standard. The parallel interface uses 8 or 10 bits of multiplexed YCbCr, chroma and luminance data, and 27 MHZ clock with embedded unique timing codes or external video timing signals for BT.656 and BT.601 respectively. Sometimes, for applications like LPR, only gray scale images are required in processing.

1.2.2 Memory and Data movement

Efficient memory usage and transferring is an important consideration in a real time system design. The active or full video data should be transferred to preprocessing backend device by a parallel or serial interface via a DMA transfer from a performance standpoint. In PC world, I/O cards connected to PC interconnected buses transparently do that process with help of operating system drivers. In an embedded design, integrated system blocks like interfaces intelligently decoding preamble codes or memory segmentations to reduce processor core operations are considered in low level design.

1.2.3 Preprocessing

In dealing with plate localization or character segmentation, initial processing of raw image to correct distortions, eliminate noise present in the data may be required before real analysis of image sequences. These operations may include any or a combination of followings; digital convolution, point operations (contrast enhancement, threshold), global operations, neighborhood operations (smoothing or sharpening), geometric operations, or temporal operations. A particular image enhancement method selected due to specific requirements like noise reduction or contrast enhancement of input images would help to yield the best recognition performance in next steps.

1.2.4 Plate Localization

It is a complex task in a real-time environment as well as an important stage in LPR system. Assuming a correct localization of a plate despite lots of fake ones in frame, the overall system is reduced to a typical recognition system.

Generally, in edge and texture based methods, LP localization task is completed in two major steps as candidate extraction and candidate verification [1].

Candidate extraction may be accepted as a straightforward issue out of account verification process which is intractable results in lots of situations. This leads to a certain amount of trial and errors to find and use a consistent extraction method of locating correct regions.

A number of factors getting process complicated in field may be listed as following; the variety of size and colors, complexity of background, weak or strong illumination, changing weather conditions, variations in shape, dust on plate, fake objects (radiators, screws, texts), natural scenes, blurring or other deformations during image acquisition etc.

1.3 Related Works

License Plate Localization is a key and challenging issue since it is distinctive part of LPR systems that finally aim to recognize plate numbers by usually using well known and mostly standard character recognition algorithms. Once the location of the license plate is found, it is reduced to a recognition stage. That means correct detection of plate location which is the most difficult part because of various undesired conditions as given previous section. It affects the overall performance.

Different LP localization techniques are introduced in literature. These methods are generally a combination or one of edge detection based, plate location color or texture based approaches.

Hsieh and Yu [2] use morphological methods for detection of license plates in complex scenes. Hongliang and Changping [3] proposed a method based a mixture of edge statistics and morphology. Kamat and Ganesan use hough transform for line detection module [4]. Hough Transform is used in other work in addition to voted block matching for the extraction and tracking of license plates [5]. Kahraman and Gokmen uses the Gabor filter in detection phase as edge detection and threshold are proposed for binary imaging of gray level images [6]. A Threshold function and template matching are used by Yohimori and Mitsukura for their detection process [7]. Shapiro and Gluhchey [8] use a combination of edge detection and vertical pixel projection to detect location. Dlagnekov and Belongie use the normalized cross correlation [9].

Those studies mentioned above have not been specifically designed for embedded systems except Kamat and Ganesan implemented Hough transform based method in DSP architecture [4]. An FPGA is used by Kang and others [10] and Bellas and others [11]. The system is implemented on an embedded DSP platform and processes a video stream in real-time with an AdaBoost based approach in detection stage [12]. An FPGA based Plate Recognition System which is portable and faster than computer based systems is researched by using Gabor filters, Threshold and Connected Component Labeling [13].

In recent years, it is challenging to use embedded systems in different kind of application areas of vehicle systems like remote traffic surveillance, vehicle detection, vehicle plate detection, vehicle speed estimation [14-[17]. It satisfies low cost, compactness and efficiency constraints in real time video applications.

1.4 Proposed System Architecture

The embedded system architecture which is proposed in this work is to stimulate system and design specialization in addition to algorithm enhancements for an edge detection based LP localization algorithm similar to one proposed by Beratoglu [18]. The target system is differently an embedded platform which is an Analog Design BF651 dual core fixed point multi instruction DSP based evaluation kit ideally suited for a wide range of video surveillance systems.

The software system design is a hard issue spanning from board bring up to assembly code optimizations with multi instructions, hardware loops in terms of real time constraints. The system presented is generally designed for LP localization. However it may also be a part of any general surveillance system for future works.

A sample of overall system scenario is plotted in Figure 1.1 for a license plate recognition application on a PC based system developed by ITU Software Development Department which is a product upon several academic studies and DIVIT software development.

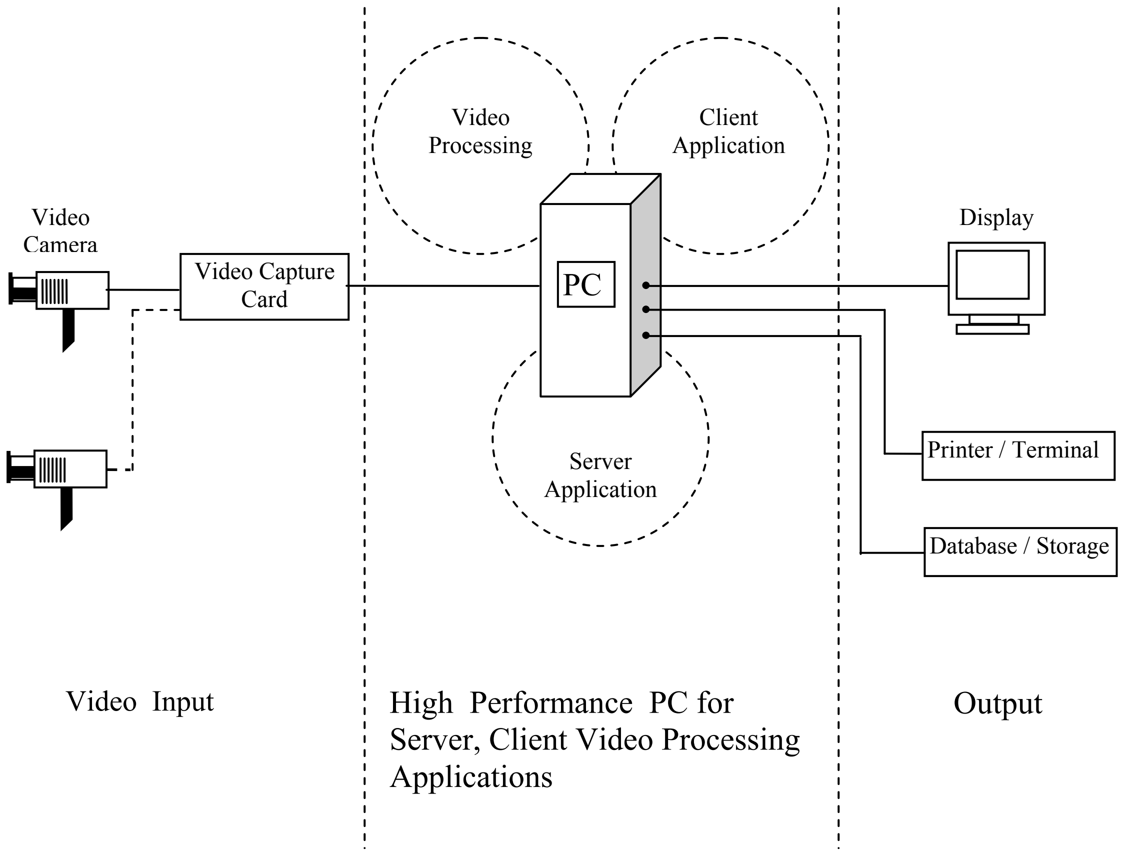


Figure 1.1 : A License Plate Recognition Application on a PC Based System

The sample block diagram of an embedded DSP based LPR system is given at Figure 1.2 and Figure 1.3 suggested replacing the system above. The first one consists of a number of embedded LP localization systems that are remotely controlled by a master operation unit. The plate regions detected in input image is segmented and transferred to recognition module and data storage media of master unit if required.

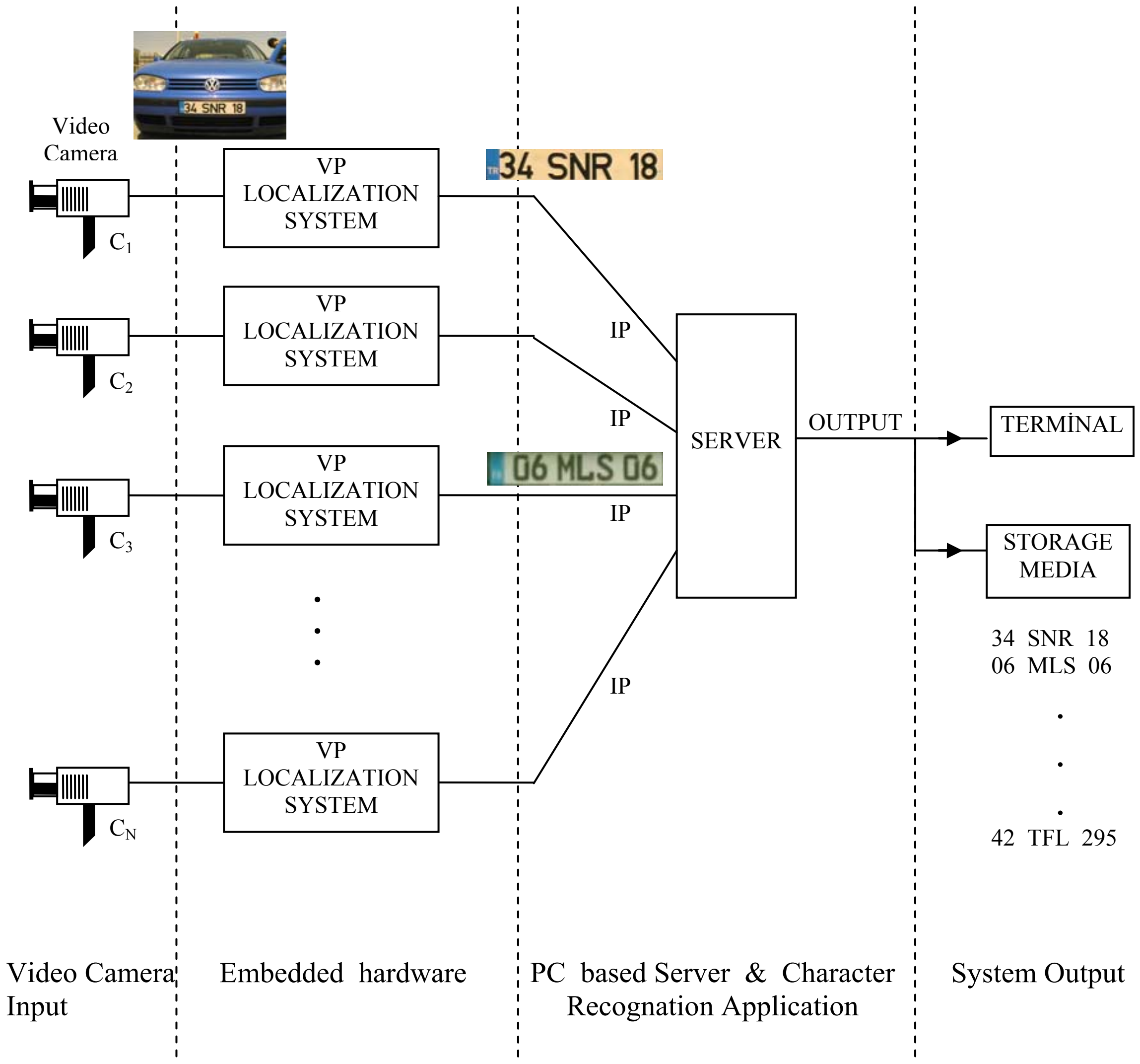


Figure 1.2 : DSP Based LPR System - 1

The second system is thought to be stand-alone working device directly outputting results through a terminal or storage media without any external operation unit. In this system, the recognition and the localization processes are done in the same embedded unit. It may be either connected to a server unit for operations like custom controls of processing and database management of real-time inputs and outputs.

Both of these systems are thought to have advantages in terms of embedded hardware simplification, portability, compactness, cost reduction of design as compared with PC based system.

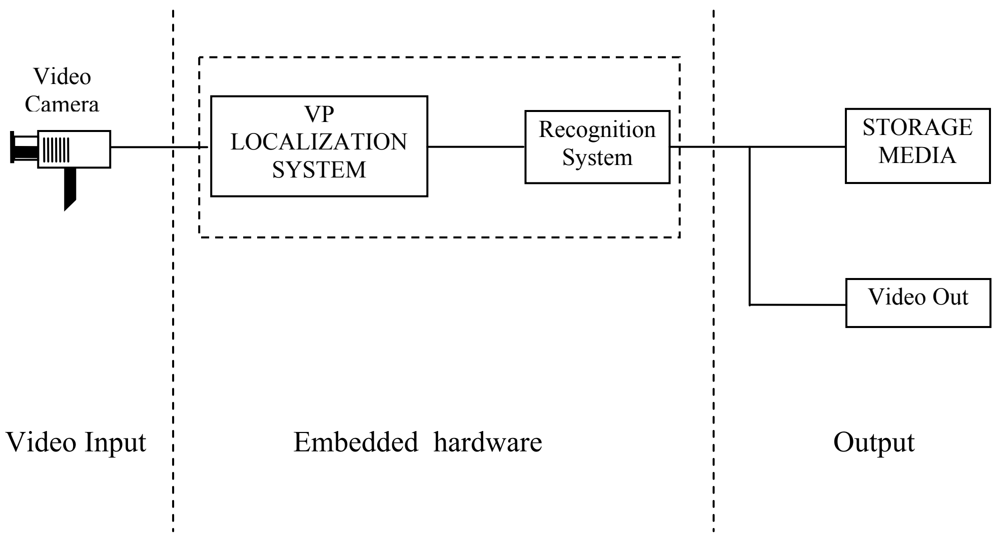


Figure 1.3 : DSP Based LPR System - 2

1.4.1 Hardware Description

Basic hardware modules of embedded system design given at Figure 1.4 are described in below sections. There is video decoder at the front-end and video encoder at the backend of video processor connected through high speed parallel ports. The input port has configurable section filters to parse D1 video data. The flash and external memories are connected to processor through local memory interface.

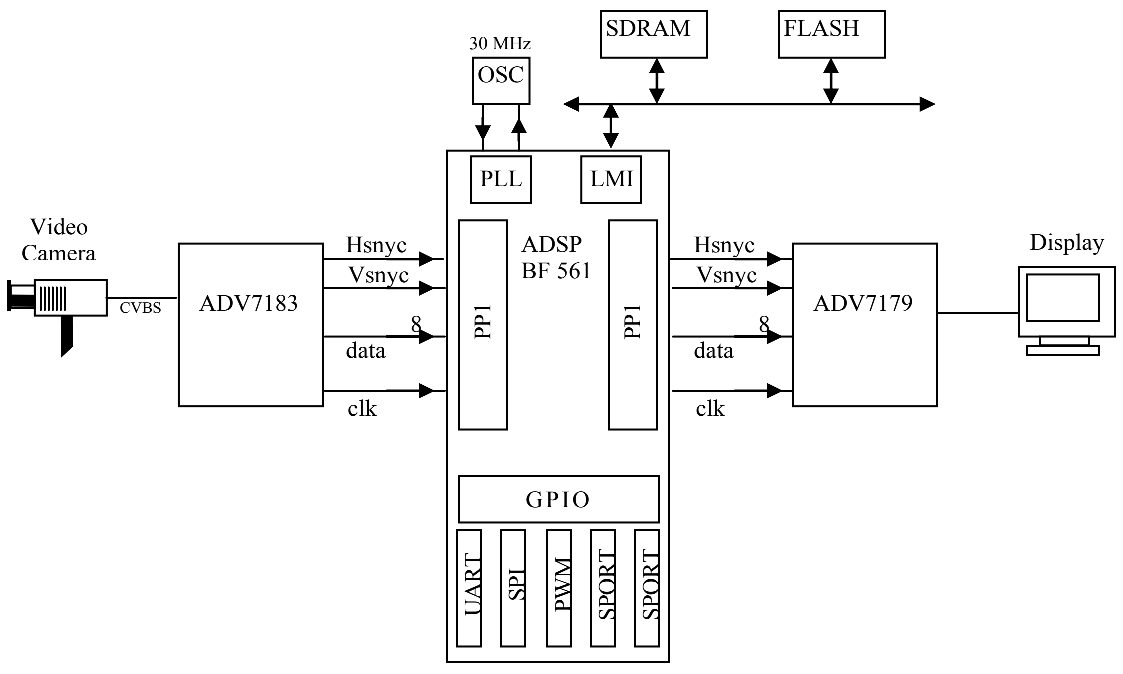


Figure 1.4 : Hardware Design Block Diagram

The interconnect architecture of Blackfin ADSP is given at Figure 1.5 [19]. There are three types of memory in the system due to Enhanced Harward Architecture. While internal L1 type memory works within single CPU cycle, L2 type latency is slower than L1 type memory even though its size is large. A number of programmable IRQ which generates events to callback functions are available. An important application in our situation is the event handling mechanism established to be interrupted with DMA controller signal after each video field received from decoder.

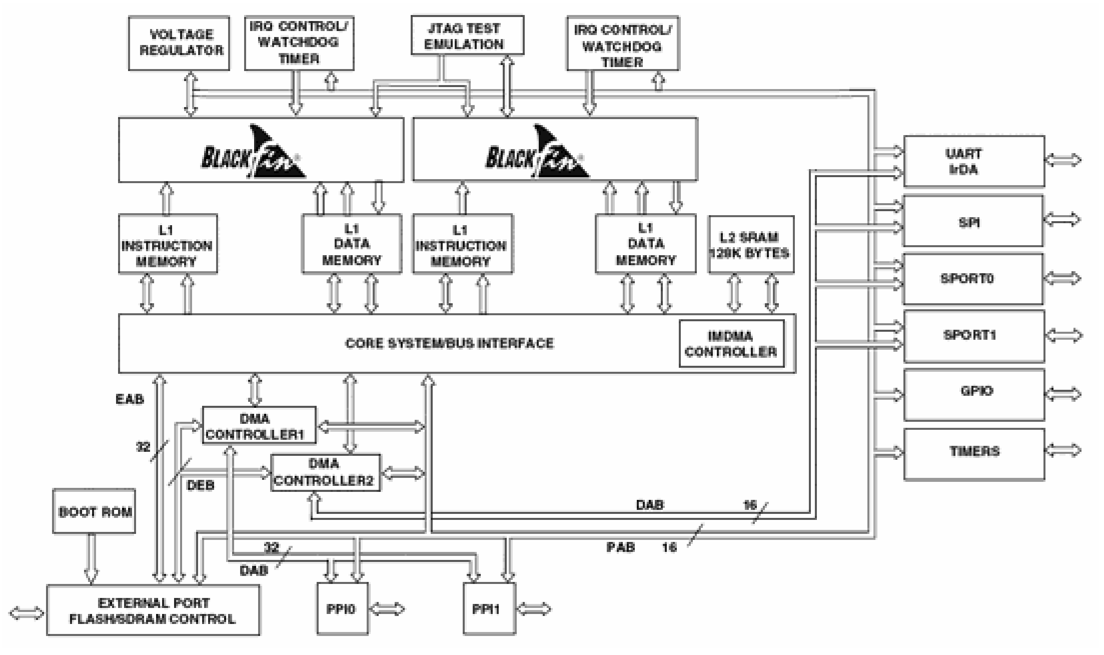
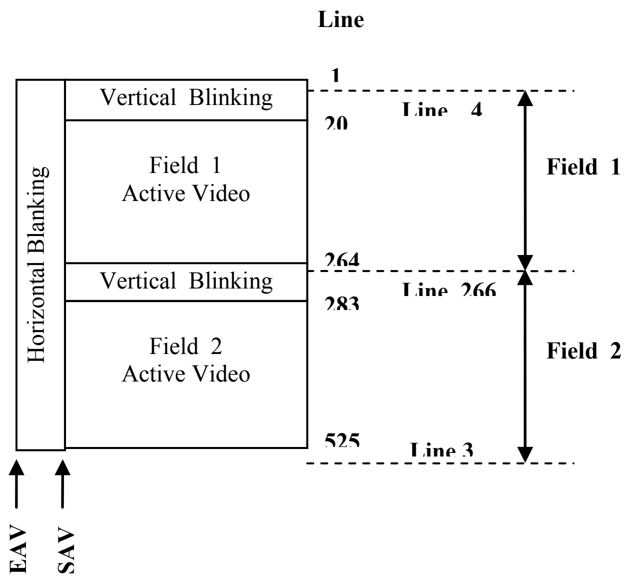


Figure 1.5 : Internal DSP Architecture

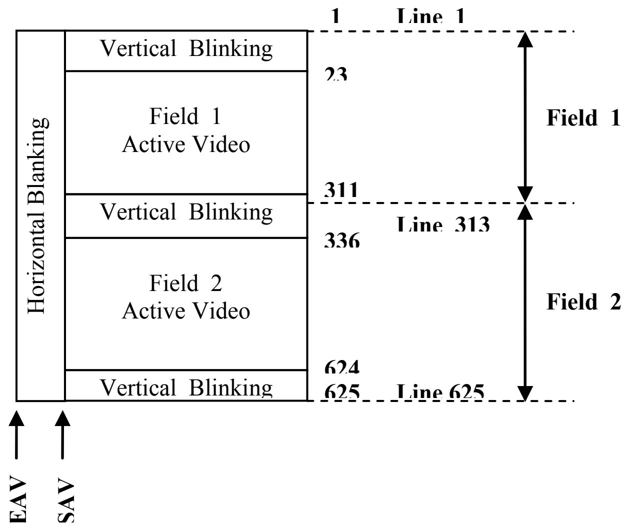
1.4.1.1 Video Acquisition

Base band CVBS or S-VIDEO signals are typical standard outputs of analog video cameras. A video decoder chip is required to convert this analog signal to a digital stream. This decoder is a peripheral device controlled with I2C communication by master device which is video processor. There are number of similar chroma decoder devices in the market. These devices are generally programmed with corresponding configuration settings for any of PAL or NTSC TV systems through I2C communication.

For a standard definition video input, the active and blank video intervals for NTSC and PAL systems are given respectively in Figure 1.6. It illustrates picture resolution in odd and even fields in terms of active video and blanking intervals locations in raster scan.



Line Number	F	V	H (EAV)	H (SAV)
1 – 3, 266 - 282	1	1	1	0
4 -19, 264 - 265	0	1	1	0
20 - 263	0	0	1	0
283 - 525	1	0	1	0



Line Number	F	V	H (EAV)	H (SAV)
1 – 22, 311 - 312	0	1	1	0
23 - 310	0	0	1	0
313 – 335, 624 - 625	1	1	1	0
336 - 623	1	0	1	0

Figure 1.6 : ITU BT.601/656 Video Data Structure

This stream may be ITU BT.601/656 PAL or NTSC signal in general depending on application field. PPI1 which is generic high rate data input of processor is used to be connected to digital video output of video decoder device. The output of video decoder is an 8 bit parallel stream with pixel clock in addition to embedded SAV/EAV codes or external signals for ITU.BT656 and ITU.BT601 respectively. Real Time Stream is parsed at video processor to get active video of fields utilizing embedded synchronization codes or external signaling. Preferably all active and vertical sections may be considered to be parsed, and then manually filtered.

It is illustrated as digital stream for PAL system in Figure 1.7. Pixel clock is 27 Mhz for a standard definition resolution. Chroma and luminance data are streamed in given order for each pixel in a line.

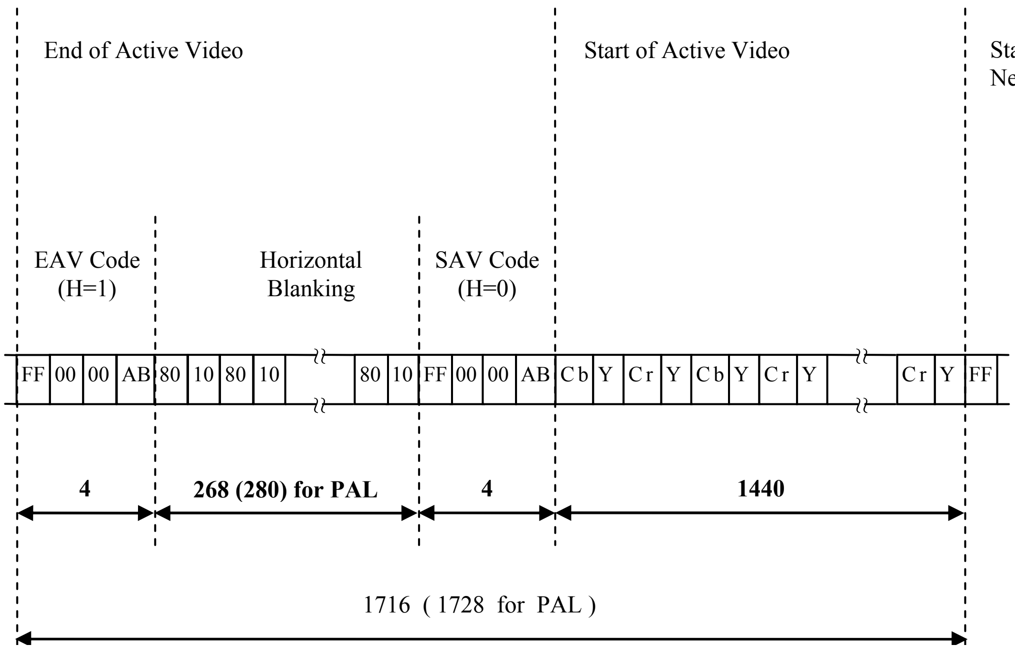


Figure 1.7 : ITU BT.601/656 Serial Stream

1.4.1.2 Video Data Movement

Active part of D1 type digital video stream should be parsed relative to synch signals and captured in system memory in a combination of memory, DMA and interrupt management. Video Processing Processors usually have multiple, independent DMA controllers that support automated data transfers with minimal overhead form processor core. Video data should be transferred to a safe buffer before processing in terms of real time constraints. This is a part of input video frame or a number of macro blocks depending on application requirements. At the end of each video line or frame, an interrupt is configured to generate a callback for video processing application to be semaphored. The DMA copies video from external memory or peripherals to internal memory if available via 2D DMA engine for best performance. Interrupt service routines to signal callbacks when video data is ready for processing purposes.

The generic 2D to 1D data movements used for system development are illustrated in Figure 1.8. The sampled digital video is preferably double buffered. A part of input image is captured for processing purposes and moved to another buffer configured. As seen on figure, DMA destination and source parameters for the number of pixels and number of lines to be replaced are configured. The number of pixels to be copied in a line is determined with a increment of X_{modify} pixels at each DMA iteration until X_{count} reached. Similarly Y_{modify} and Y_{count} configures the number of lines to be replaced in a frame.

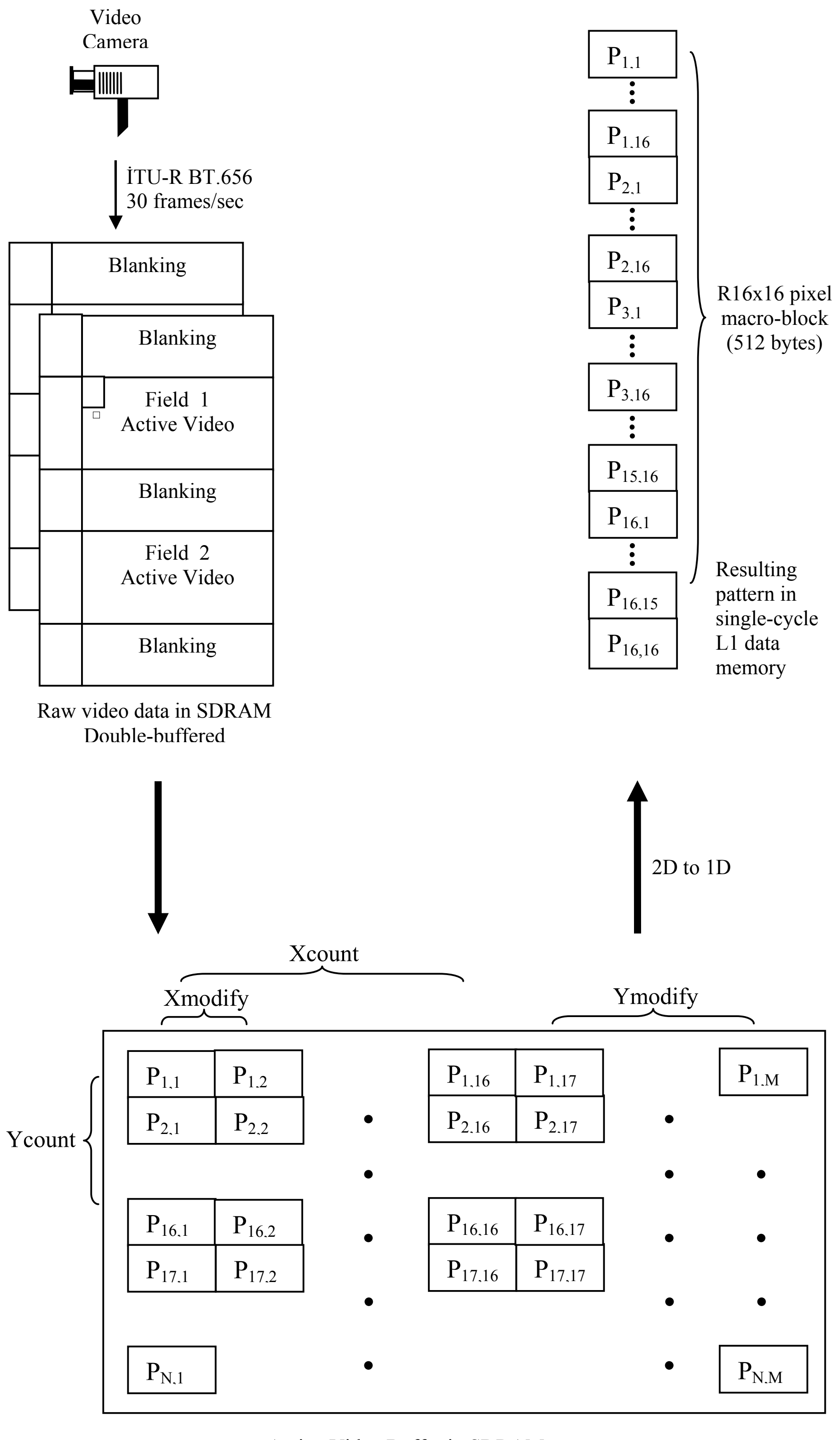


Figure 1.8 : 2D to 1D DMA Access

1.4.1.3 Internal and External Memories

The LP Localization algorithm is processed on Level1 (L1) and Level2 (L2) memory blocks which are a part of modified harward architecture in a combination with a hierarchical memory structure. L1 memory is connected directly to processor core, runs at full system clock speed, and offers maximum system performance for time critical algorithm segments. L2 memory is larger, less performance than L1 type memory, but faster than off-chip memory. SDRAM is used as external off-chip memory which is called as L3 type. Memory model is critical at system performance as well as assembly code optimizations.

1.4.1.4 License Plate Localization

Video processing pixel operations are done by using assembly instructions for consistent optimizations due to multiple pixels processing in a single clock.

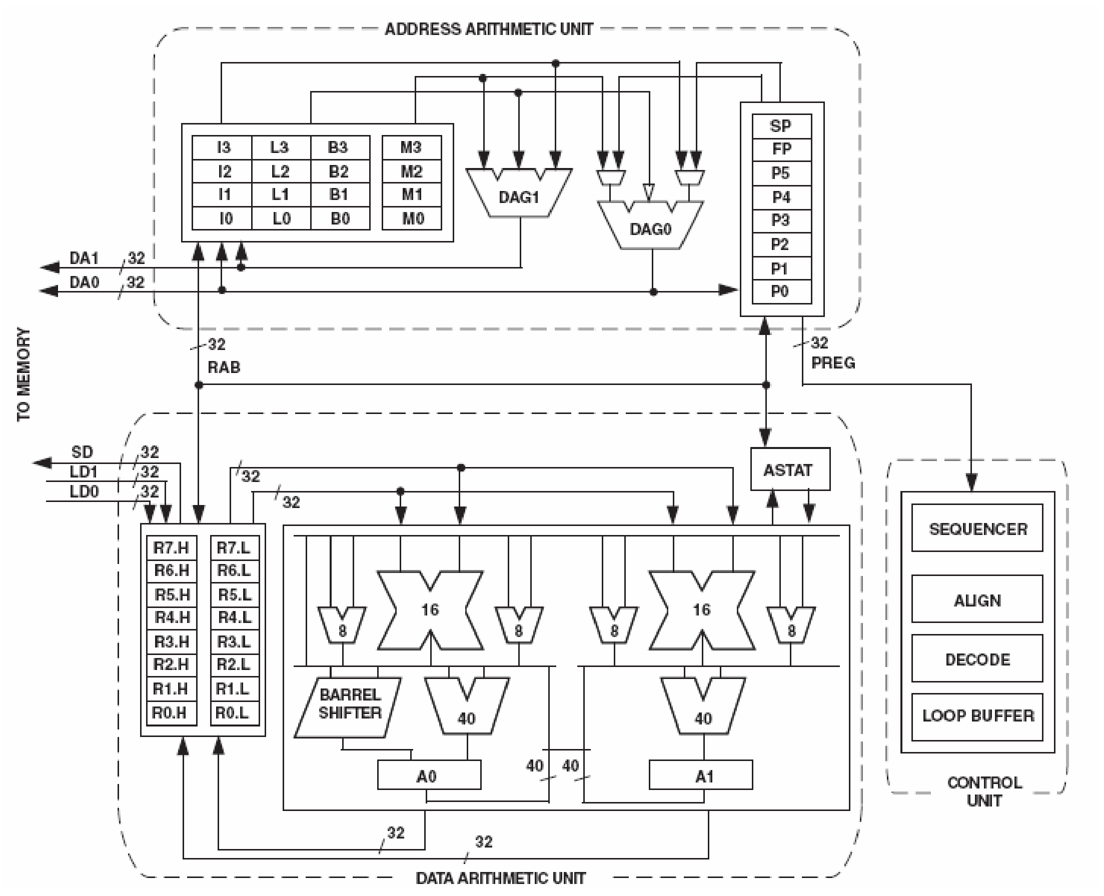


Figure 1.9 : ALU, DAU, and Control Unit

As shown in Figure 1.9, each Blackfin core contains dual 16x16 multiplier/accumulators (MACs) , dual 40-bit ALUs performing 16-/32-/40-bit operations, video ALUs, and a single shifter [19]. There are 8x32 bit registers and

2x40 bit accumulators in the data register file used to hold 32-bit values or packed 16-bit values. Only one core is used in our case. However, the second one may be configured in a parallel processing architecture by setting up a synchronization mechanism between two processors. Two data address generators (DAGs) provide addressed for simultaneous dual operand fetches from memory. Four sets of 32-bit Index, Modify, Length, and Base registers are shared. Eight additional 32-bit pointer registers is used for general indexing of variables and stack locations.

The system processor is a member of Blackfin family which is a fixed point DSP supporting fractional and integer data formats. Arithmetic operations are done in a fractional binary format denoted by 1.15. The floating point operations may be simulated in software but not efficiently. However, the calculations are applied in a normalized area in the 1.15 format, 1 sign bit and 15 fractional bits represents values from -1 up to 0.999969.

Table 1.1 : Bit weighting for 1.15 Fractional Data Types

2^{0}	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}	2^{-9}	2^{-10}	2^{-11}	2^{-12}	2^{-13}	2^{-14}	2^{-15}

Table 1.1 shows the bit weighting for 1.15 numbers as well as some examples of 1.15 fractional numbers and their decimal equivalents in Table 1.2. The first bit of the fractional data type is sign bit. Unless given any operation mode, all arithmetic and multiplication operations are done in fractional mode and gives results normalized in decimal equivalents between -1.000000 and 0.999969.

Table 1.2 : Some of 1.15 fractional numbers and their decimal equivalents

1.15 NUMBER (HEX)	DECIMAL EQUIVALENT
0x0001	0.000031
0x7FFF	0.999969
0xFFFF	-0.000031
0x8000	-1.000000

During the same cycle, multiple data fetches occur in parallel with MAC operation. This method allows efficient computation of two output point in loop operations of convolution and makes spent time reduce by half during operations of calculating gradients and find edge points in first stage of vehicle plate localization process.

Once the localization process is completed, the corresponding information is transferred to output stage.

1.4.1.5 System Output

An output of the LPR system may be in general one or a combination of video output device or reports screen through a terminal or storage media device. PPI2 general purpose high speed 8-bit parallel output of blackfin processor is used as ITU BT.656/601 based video output to interfacing with video encoder device if direct video output is required.

1.4.2 Software Environment

The embedded development tools for Analog Device Blackfin processors which are integrated to VisualDSP++ tool chain including assemblers, C/C++ compilers, linkers, loaders and other utilities like statistical profiling tool are used during design development.

These tools are working on a hardware reference design which picture is given at Figure 1.10. USB and JTAG is used for diagnostic control unit interfacing, debugging, and flashing purposes.

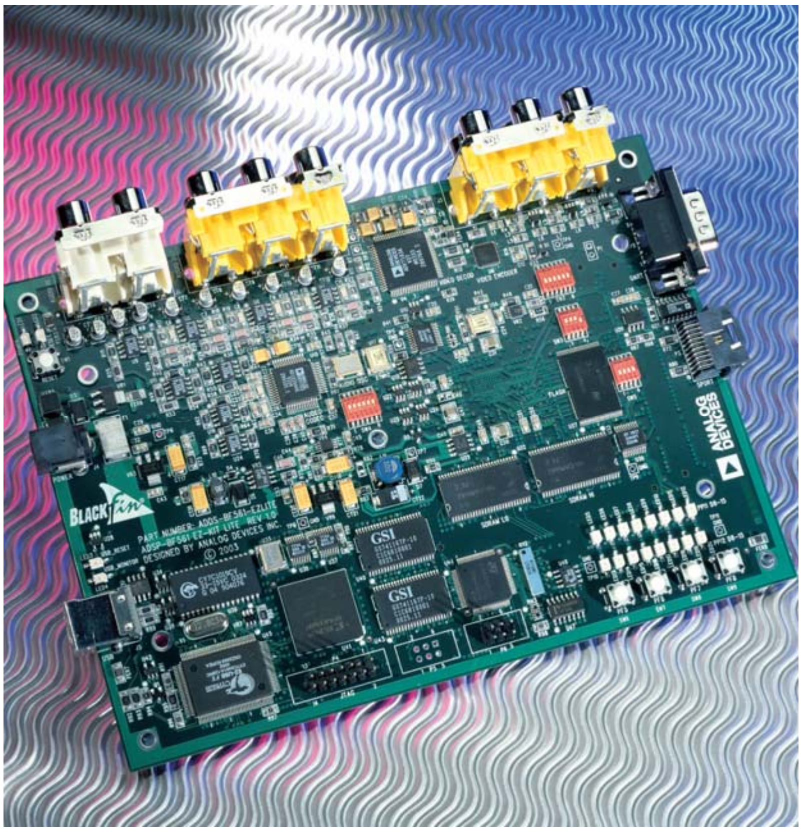


Figure 1.10 : BF561 EzKit

The linux development environment is also evaluated by the reason that it is highly capable of connectivity features. However it is not actively used during development process since the main output for demonstration purposes in this work is thought to be video output.

Additionally MATLAB scientific tool is also used for algorithm development, evaluation and verification period in conjunction with VisualDSP++ tool chain.

1.4.3 Advantages of the System Proposed

The architectures minimizing the system cost of target application is a critical issue in computer engineering today. Much more powerful CPUs are required in the market for specific video applications. The specific devices instead of general CPU architecture and system design may be stimulated to satisfy two major key constraints which are performance and cost. The advantages of DSP based video application architecture may be summarized as following.

- Integrated system design
- Powerful and flexible memory access architecture
- High Speed Video Streaming Ports
- Embedded Active Video / Blank Interval Filtering
- Specific Video and Pixel Processing Instructions
- Parallel Instructions, Dual Data Fetches, Hardware Loops
- Cost reduction

On the other hand, embedded systems are so complicated and development time consuming devices especially in real-time environments that more experience is required than general software architectures.

2. LICENSE PLATE LOCALIZATION

In the following sections, all the details of the algorithms and embedded implementation of the license plate localization system are given. Firstly the introduction of the reference system is denoted.

2.1 Reference System

The algorithms used in this work are generally similar to one presented in [18]. These algorithms are successful in terms of academic and field proven results by several recognition systems developed by DIVIT and ITU Software Development departments [18],[20]. Beyond these algorithms, in some stages, some enhancements and modifications are suggested like done in blob coloring and plate and character candidate detecting rules. It is also aimed to design a software structure in consistent with an embedded video surveillance platform which propose low power consumption, compact design, high performance and low cost.

2.2 Overview

The algorithms may be reviewed in three stages. The first phase is edge detection based segmentation of objects in image. The theory of Canny[21] are firstly described in addition to given efficient embedded implementation of calculation of gradient images and discarding adjacent non-extreme values of pixels of gradient image.

In the following section, the threshold operation based on detected edge pairs due to negative and positive magnitudes in a sequence through lines and connected component labeling details are given.

At last two sections, a number of rules are method to correctly find the features of plate character candidates and plate locations are introduced.

2.3 Segmentation

Edge points are sharp variations which are related to a boundary between other objects or background scene. The edge points may be used to find the locations, and the area that it belongs to and other features of the objects.

In our case given input images consisting vehicle, it is observed that the frequency of variations are increased in some regions. We know that the plate region must have edge points related to sharp changes through plate background, number and other characters. In other ways, the plate region is one of the sources of sharp variations in input image. It should be mentioned that some of other regions may have similar attitudes in terms of edge features. In this point, it is obviously visible that our next task must be to find additional rules that should be determined to distinguish plate regions from other similar areas [18].

2.3.1 Canny Edge Detection

There are lots of edge detection approaches giving different results in particular situations. As a part of this work, Canny Edge Detection algorithm is selected and optimized for embedded processor hardware after a certain amount of trials. It is typically consist of enhancement, non maximum suppression, and hysteresis stages. Since the system processor is a fixed point DSP, the calculations are done with fractional data types. The input 8-bit image data are converted to consistent 16-bit fractional data.

The theory of Canny [21] is based on three major optimality criteria of performance which are good detection, good localization, single response constraints.

2.3.1.1 Good Detection

The optimal detector must find real edges meanwhile do not response to fake edges. This is achieved by a filter maximizing the signal to noise ration (SNR) assuming that a degraded ideal step edge with noise. The response of a linear filter, f, to this edge is

$$\int_{-w}^{w} G(x) f(x) dx = A \int_{-w}^{0} f(x) dx \qquad \textbf{(2.1)}$$

The good detection criterion becomes

$$SNR = \frac{A\left\| \int_{-w}^{0} f(x)dx \right\|}{n_0 \sqrt{\int_{-w}^{+w} f^2(x)dx}} \tag{2.2}$$

Where n_o is RMS noise amplitude per unit length.

2.3.1.2 Good Localization

The location of found edges should be matched to correct edges as much as possible. The RMS distance between detected edges and true edges gives a criterion for correct localization.

$$LOC = \frac{A\left\| f'(0) \right\|}{n_0 \sqrt{\int_{-w}^{w} f'^2(x)dx}} \tag{2.3}$$

2.3.1.3 One response to one edge

Canny also defined additional criteria which guaranties one response to one edge.

$$T = \pi \left[\frac{\int_{-\infty}^{+\infty} h^2(x)dx}{\int_{-\infty}^{+\infty} h'^2(x)dx} \right]^{1/2} \tag{2.4}$$

2.3.2 Gradient Image

Concerning the first two criteria, Canny has shown that a very good approximation of edge detector is the first derivative of Gaussian. The function of a 2D Gaussian signal and its first derivatives are given in equation (2.6) and equation (2.7) respectively.

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{2.5}$$

Instead of applying Gaussian smoothing to input and then computing the gradient components, the image is convolved with the first derivative of Gaussian directly.

Since Gaussian function is linearly separable, the calculation may be done independently for both directions in terms of computational efficiency.

$$\frac{\partial G(x,y)}{\partial x} \propto xe^{-\frac{x^2+y^2}{2\sigma^2}} \quad \textbf{(2.6)}$$

$$\frac{\partial G(x,y)}{\partial y} \propto ye^{-\frac{x^2+y^2}{2\sigma^2}} \quad \textbf{(2.7)}$$

In our case, only the horizontal gradients are enough to find edge pairs within a restricted neighborhood through a scan line due to chancing negative to positive gradient magnitudes.

2.3.2.1 Calculation of Gradients

The kernel coefficients are calculated from the differentiation of Gaussian function to convolve the input image. The number of coefficients is restricted within a window size in consistent with the sigma value selected. The size of window is determined due to negligible coefficients of Gaussian Function. The coefficients for mean and sigma values of Gaussian function equal to zero and 2 respectively are calculated as denoted in Table 2.1. The window length selected is five.

Table 2.1 : Floating Number of Differentiation of Gaussian Coefficients

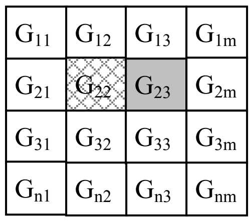
0.0021	0.4258	0.0000	-0.4258	-0.0021

The floating point operations take very long time on BF561 since it is a fixed point processor. The convolution process is done in fractional data formats efficiently. A correct data conversion should be applied to input image data and filter coefficients. Firstly the coefficients of filter are converted to 16-bit fractional data as given at Table 2.2

Table 2.2 : Fractional Data of Differentiation of Gaussian Coefficients

0x0044	0x3680	0x0000	0xC97F	0xFFBB

By aligning the input data properly, both Blackfin multiply accumulate (MAC) units can be used in a single processor cycle to process two output points at a time during filtering. During this same cycle, multiple data fetches occur in parallel with the MAC operation. This method shown in Figure 2.1 allows efficient computation of two output points for any iteration of the loop. In other words, it takes 4.5 cycles per pixel instead of the 9 cycles per pixel [22].



G_{11} G_{12} G_{13} G_{1m}
G_{21} G_{22} G_{23} G_{2m}
G_{31} G_{32} G_{33} G_{3m}
G_{n1} G_{n2} G_{n3} G_{nm}

g(x,y) Output Matrix

▨ $G_{22} = H_{11} \times F_{11} + H_{12} \times F_{12} + H_{13} \times F_{13} + H_{21} \times F_{21} + H_{22} \times F_{22} + H_{23} \times F_{23} + H_{31} \times F_{31} + H_{32} \times F_{32} + H_{33} \times F_{33}$

■ $G_{23} = H_{11} \times F_{12} + H_{12} \times F_{13} + H_{13} \times F_{14} + H_{21} \times F_{22} + H_{22} \times F_{23} + H_{23} \times F_{24} + H_{31} \times F_{32} + H_{32} \times F_{33} + H_{33} \times F_{34}$

Loop Cycle **(F11 Loaded into R0.H, F12 Loaded into R0.L, H11 Loaded into R1.L)**

```
1   A1 = R0.H * R1.L,     A0 = R0.L * R1.L      || R0.L = w[I0++] || R2 = [I3++];   //I0-I3 are index regs
2   A1 += R0.L * R1.H,    A0 += R0.H * R1.H     || R0.H = w[I0 - -];                // w[] is 16-bit acces
3   A1 += R0.H * R2.L,    A0 += R0.L * R2.L     || R0 = [I1++]    || R3 = [I3++];
4   A1 += RO.H * R2.H,    A0 += R0.L * R2.H     || R0.L = w[I1++];
5   A1 += R0.L * R3.L,    A0 += R0.H * R3.L     || R0.H = w[I1- -]    || R1 = [I3++];
6   A1 += R0.H * R3.H,    A0 += R0.L * R3.H     || R0 = [I2++];
7   A1 += R0.H * R1.L,    A0 += R0.L * R1.L     || R0.L = w[I2 ++]    || R2 = [I3 ++];
8   A1 += R0.L * R1.H,    A0 += R0.H * R1.H     || R0.H = w[I2 - -]   || R1 = [I3 ++];
9     R6.H = (A1 += R0.H * R2.L),   R6.L = (A0 += R0.L * R2.L);            //Accumulate for 2 outputs
```

Each time through this loop yields two output points
Total Cycles = 9 for every 2 pixels => 4.5 cycles per pixel

Figure 2.1 : Convolution Efficiency Sample for 8-bit input valued input buffer with 3x3 masking on DSP Architecture

The optimizations in convolution like any other algorithm of this system are basically done by utilizing three nested features of the system which are variable instruction lengths, dual MAC operations, or dual ALU operations. It is possible to access 16/32/64- bit instruction operational codes directly in parallel with a number of restriction rules to obtain the best code density while maintaining high performance. The first assembly line in Figure 2.1 consists of an 32-bit instruction in addition to two 16-bit length data fetch instruction in parallel. The total number of instruction length is restricted to 64-bit. The first 32-bit instruction which may be an example of dual MAC operations uses two accumulators, two DAG and two MACs to accumulate A1 and A2 with fetched values from upper and lower sections of data registers R0 and R1. It therefore doubles the MAC throughput. All these operations are done in a single clock cycle. ALU operations support a number of executions in a single cycle. This may include one of single 16-bit operation, dual 16-bit operation, quad 16-bit operation, single 32-bit operation or dual 32-bit operation as giving examples below.

Table 2.3 Algorithm Optimizations with ALU operations

Mode	Example Instruction
Single 16-bit operation	R6.H = R3.H + R2.L
Dual 16-bit operation	R6 = R2 +|– R3
Quad 16-bit operation	R3 = R0 +|– R1, R2 = R0 +|– R1
Single 32-bit operation	R6 = R2 + R3
Dual 32-bit operation	R3 = R1 – R2, R4 = R1+ R2

As utilizing dual ALU, dual MAC and variable instruction lengths based optimizations in assembly level, it is calculated that gradient values are belonging to the magnitudes of horizontal edges.

The input image, its vertical and horizontal gradients are given at Figure 2.2 respectively.

2.3.2.2 Discarding adjacent non-extreme values

Generally the real edges may not be an ideal edge which is an instant sharp change which means the contrast value is varied reasonably over a threshold in one pixel distance. Noise or other distortions may also leads to a loss of edge sharpness which makes hard to distinguish real edge pixels from incorrect ones. Due to the fact that the edges must be distinguished from similar fakes, non-extreme values in adjacent pixels must be discarded. In this way, one response to one edge criterion is tried to be satisfied.

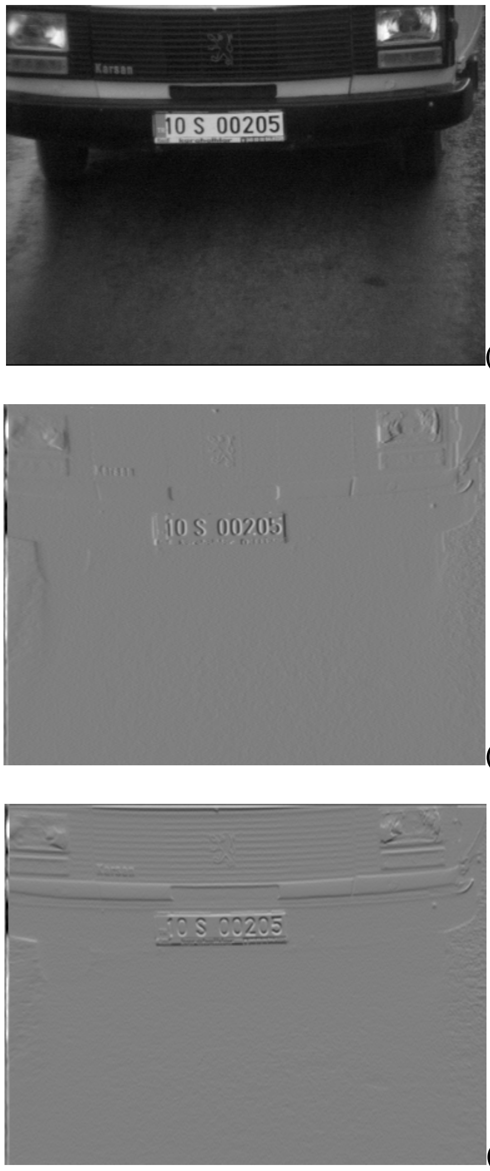


Figure 2.2 : Calculation of Gradients; (a)Input Image, (b)Horizontal Gradients, (c)Vertical Gradients

2.3.3 Threshold

The edges found as described in previous section are not directly related to plate features, but also all other objects and background features in the same image. It is tried to use sequential sharp changes in the region of plate candidates in this stage. At a glance, it may be observed that the objects in plate regions like characters and other figures borders has sharp contrast changes consecutively in a period. It is shown that the edge pairs which consist of a negative and its following positive gradient companion grouped through the image may give important information about plate location [18]. Edge companions are firstly found. The connectivity of adjacent line segments are analyzed then utilizing edge companions in order to find connected components instead of connectedness of each pixel through spatial image of edges.

2.3.3.1 Detecting Edge Pairs through a raster scan

Assuming the horizontal gradient component of an image consisting of a uniform object on a uniform background is calculated through a single horizontal raster scan, the gradient sign is changed at edge points. This means that the sign of gradient may have information about object features as well as magnitude. Utilizing negative and positive sign changes of gradients related to sharp changes in a window, we determine correctly the start and end point of an object on a uniform background.

Assuming the edge magnitude is marked as positive at the point of a transition from uniform white background to a black level region in 1D raster scan line, the magnitude will be negative at the point of transition from black to white level similarly. There will be lots of considerable gray level transitions during scanning image. However a number of them over a threshold value is only considered to discard low magnitude noise components on uniform regions.

This is one of the important steps to correctly determine license plate characters and location [18]. The features of a plate do not have constant sizes in plate region. However, in case a restriction window is predefined correctly in a scale, the objects having negative and positive gradient (start and end points) in the restricted window may be grouped together. This group may be belonging to plate characters or any other objects in frame. The window length is a parameter which depends on the image scale and may be predefined to the camera and scene setup preferences.

The algorithm for a raster line scan is given at Figure 2.3. It is applied for each line through whole image. Once a negative gradient magnitude over a predefined threshold value on a line is detected, it starts to search a positive magnitude pair. The one represented at [18] finds the pairs at first, then check the distance between pairs if it goes beyond a predefined window size. In this approach, if a negative gradient has not a positive pair in the window, it may have opportunity to lose one negative.

The negative and positive gradient pairs is applied differently that positive gradient search is discarded in case it is not found in a window to overcome this issue. Then, it starts to look for a negative gradient, and goes on.

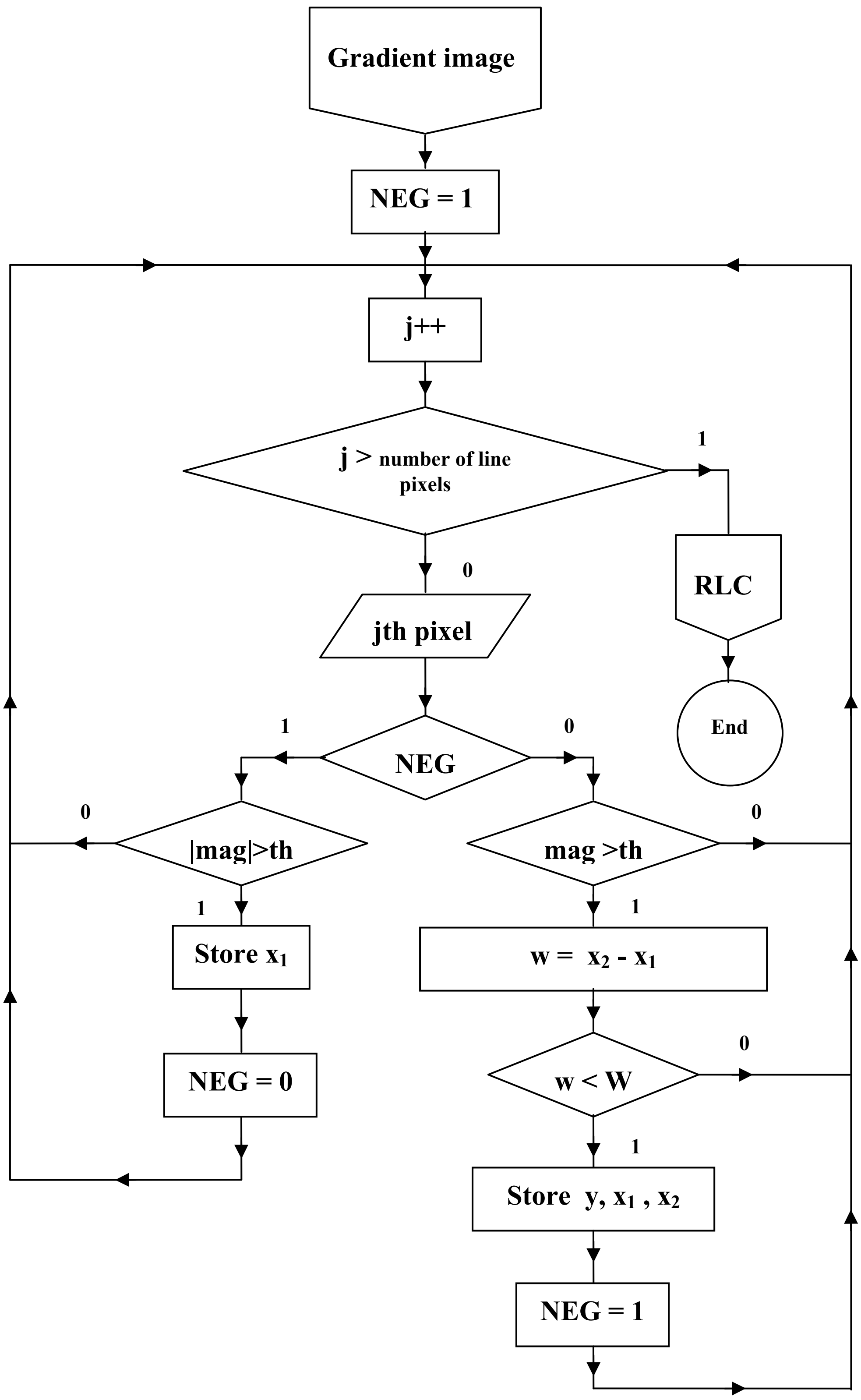


Figure 2.3 : Negative-Positive Magnitude Gradient Pairs Analysis

In fact, in real scenes, the objects are not the only source of negative and positive changes in gradient values. Because of the image degraded by noise, the image gradients may have lots of gradient sign changes. On the other hand, the magnitudes of gradients of noisy points are very low as compared to real edges. The predefined threshold value eliminates false gradient sign pairs growing out of some of noise resources.

As discarding the gradient values except the ones which are determined after gradient sign pair determination process, it is succeeded to eliminate undesirable edge points irrelevant to the issue.

The output of this system is in fact the run length codes (RLC) which are generated in order to process only interested scan lines efficiently at following stages. Instead of scanning all spatial images, a look up table of interested gradient pairs are concerned in algorithm. It means that the amount of information is reduced by compression of runs, assuming most of the spatial area is background or most of the runs are relatively long. RLC gives a high degree of sufficiency in CPU cycles which satisfies real time video constraints.

2.3.3.2 Threshold with Edge Pairs

During iterations of raster scan through whole image, the pairs are stored. Instead of using a global threshold, these data may be used for a local threshold by utilizing position of the negative and positive gradient pairs. It is the reason that the threshold term is used here and may be presented in Figure 2.4 by filling the blank pixels between each pair.

The pixel values between the coordinates of negative and positive gradient pairs through a scan line may be assumed as high white level. Similarly other pixel values will assigned as black level. As a result of that assignment, a binary image is illustrated. It gives sufficient results in terms of a right response to plate region as compared with a global threshold for a binarization process. Global threshold may not give sufficient results at low spatial resolutions, high noisy and non-uniform illumination conditions [23].

The output of the system is again not spatial image, but run length codes.

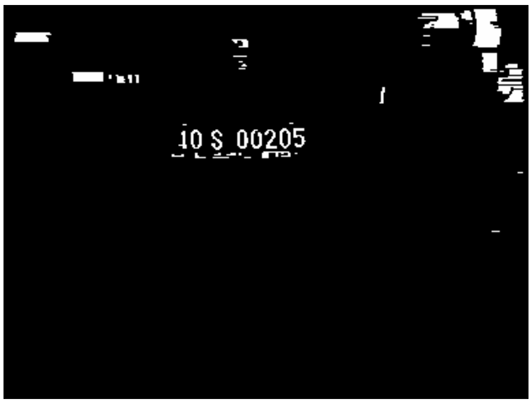


Figure 2.4 : Threshold with Negative and Positive Gradient Pairs

2.3.3.3 Connected Component Analysis

Blob coloring, or component labeling in other words, is an analysis to detect objects which are defined as a set of pixels by considering their connectedness. It is usually a point operation which the connectedness to its neighbors is classified as four pixels or eight pixels. In our work, the sections of runs are used instead of pixel processing [24]. Instead of detecting the neighborhood of pixels, the connection between runs is determined. All pixels of spatial image are not scanned. Conversely, only elements of run length codes (RLC) are considered and labeled which makes the system faster than spatial pixel based component labeling.

The algorithm in Figure 2.5 denotes this approach. It starts with first RLC element which belongs to one at top and left position in spatial coordinates. Firstly, it looks for consecutive lines. If there is no adjacent one in a neighborhood of interested line, it is discarded. The neighborhood is selected as one line here. However it may be preferably more than one line, if it is thought that the input data may have not any RLC elements complicating to find interested regions disordered because of some distortion. All bars in adjacent lines are analyzed and labeled with same tag if there is any connectivity between them.

If there is no connectivity, it is labeled with a new tag. After all adjacent lines are compared, a second pass is required which is based on label look up table updates [25] .

The labels of the connected components will be similar to sample output given at Figure 2.6

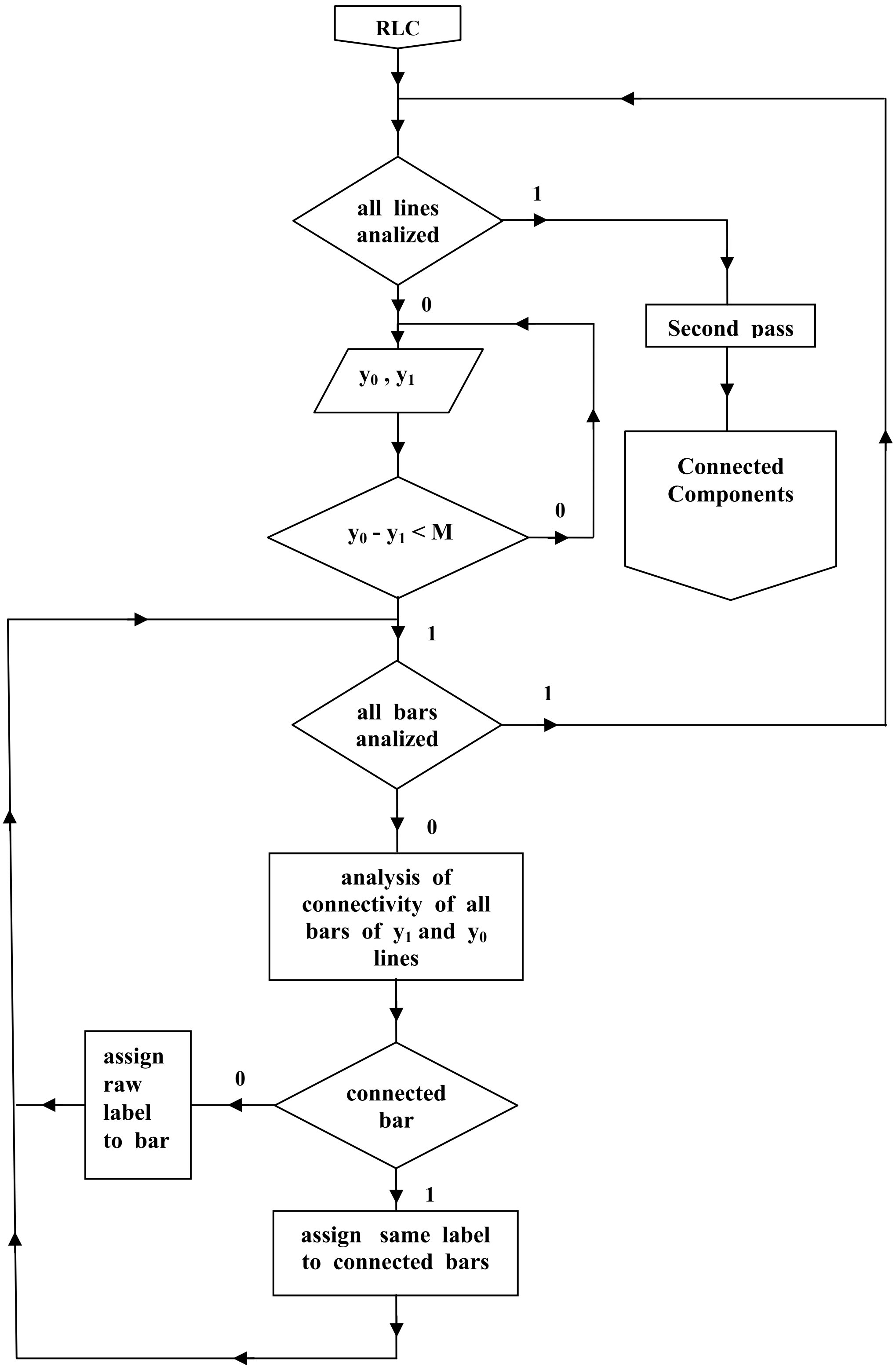


Figure 2.5 : Blob Coloring Flow Chart

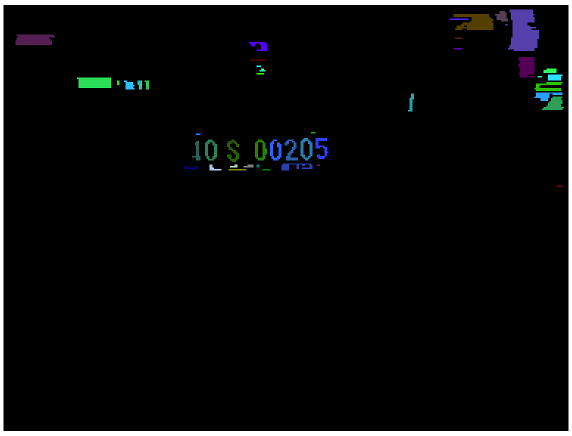


Figure 2.6 : Connected Components of Input Image

The outputs of the blob coloring process are the regions consisted of connected lines and rectangle area surrounding these regions. These rectangles will be used to determine a number of rules to distinguish plate objects (text characters) from the other objects in RLC coded image.

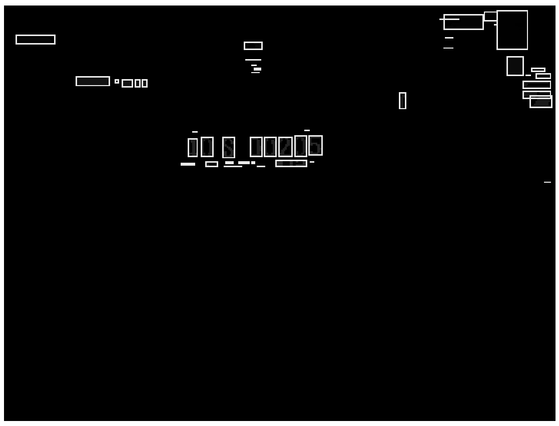


Figure 2.7 : Surrounding Rectangles of Blobs

The plate characters in Figure 2.7 are separated correctly in image. However, a character may not be separated correctly or be a combination of more than one rectangle because of disorders in adjacent lines. A merge process is defined to overcome both interconnected rectangles and non-connected rectangle problems [18]. If any rectangle is intersected, they are merged. This may be easily denoted in equation (2.8) .

$$R_i \cap R_j \Rightarrow \left[L_{maks}(R_i,R_j) < R_{min}(R_i,R_j)\right] \wedge \left[T_{maks}(R_i,R_j) < B_{min}(R_i,R_j)\right] \qquad \textbf{(2.8)}$$

The coordinates of the rectangles are compared with each others to find in case they are interconnected. Assuming that R_i , R_j are rectangles and L,R,T,B are their coordinates functions for left, right, top and bottom respectively. If L_{max} is the function that gives the highest left coordinates of two rectangles, the intersection rule for two rectangles are given as function (2.9).

Similarly, the non-connected rectangle problem may be handled to add a threshold to that function in order to merge two rectangles which the distance between them is below a predefined value.

$$R_i \cap R_j \Rightarrow \left[L_{maks}(R_i, R_j) < R_{min}(R_i, R_j)\right] \wedge \left[T_{maks}(R_i, R_j) < B_{min}(R_i, R_j) + T\right] \quad \textbf{(2.9)}$$

2.4 Detecting Plate Character Candidates

After the connected component analysis and enhancements in rectangles due to some possible disorders, it is seen in image that there are very small or big products as compared with plate characters. These fake components may be located inside plate region that makes problem more unsolvable.

It is seen that a number of rules determined to eliminate these fake characters after a trial analysis in database considered [18]. These rules may be related to plate character features like height, width, thickness. Even though all these features are variable due to different fonts, camera and scene distance, resolution and scale, it is reasonable to determine smallness, and greatness of these features that makes the problem sufficiently resolved with mentioned known conditions in a database. The width and height are considered in our work and gives sufficient results for our selected database. In different situations, it is thought that the number of these parameters may be preferably increased or decreased.

The smallness rule eliminates the noisy elements which height of a rectangle is below a threshold value. The greatness rule is similarly to eliminate the fake rectangles which height is over a threshold.

The flow chart of the algorithm is given at Figure 2.8.

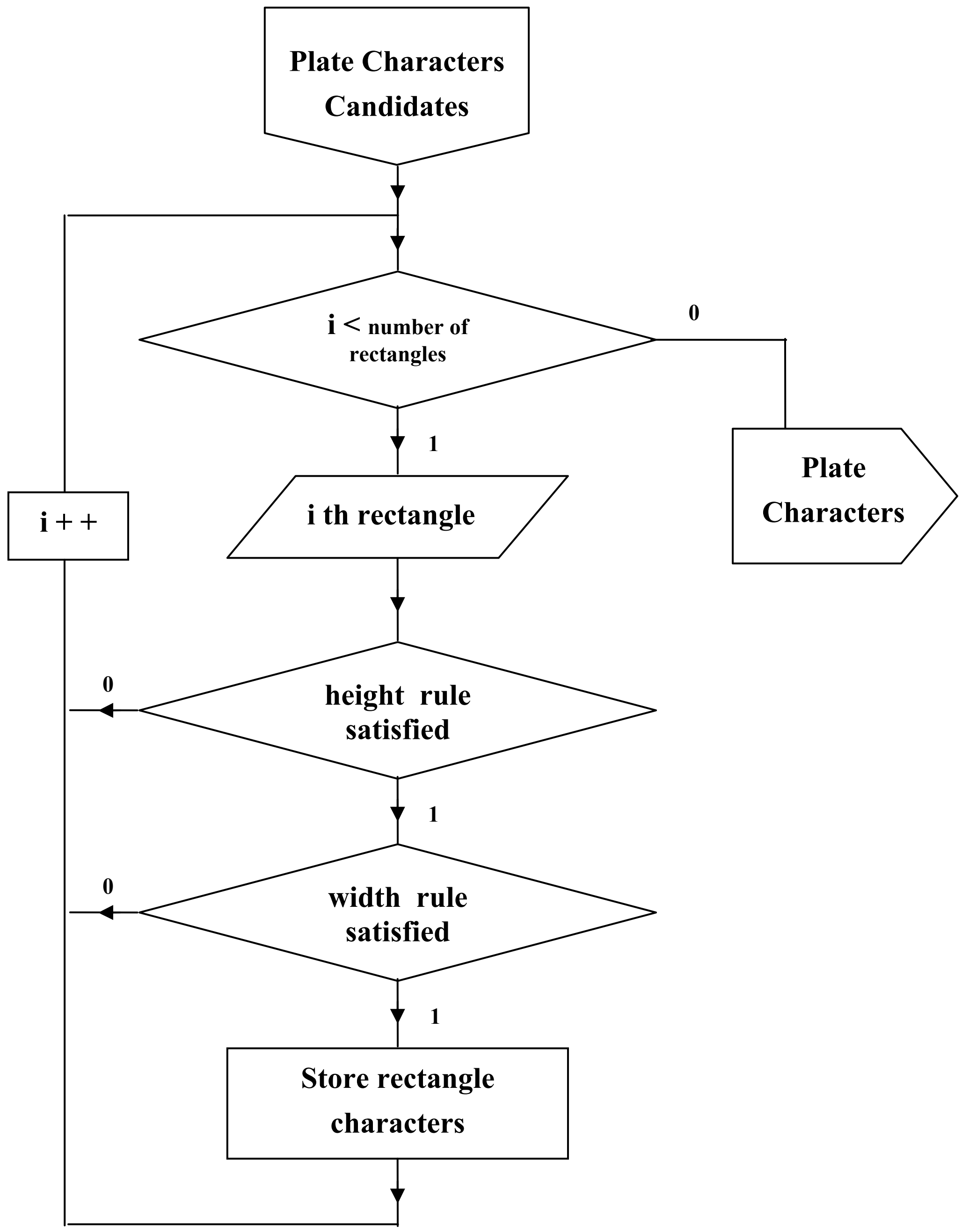


Figure 2.8 : Plate Character Candidate Detection Flow Chart

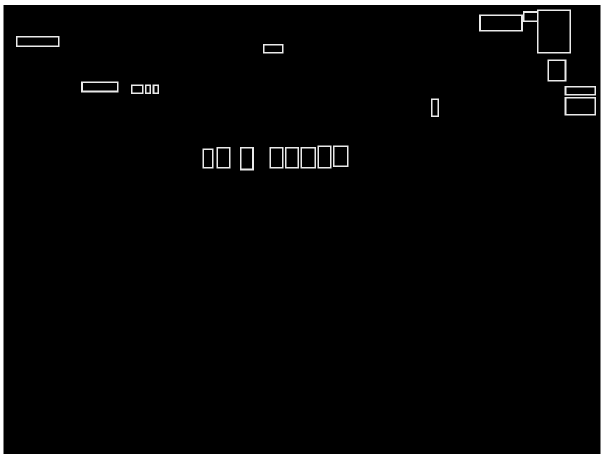


Figure 2.9 : Output of rectangles after merging and applied restriction rules

It is shown in Figure 2.9 that these two parameters are already sufficient to reduce the noisy elements on previous output of the merged rectangles section for given sample image. The image given in Figure 2.10 is similarly simplified as in Figure 2.11 by removing big and little noisy rectangles which would have been result probably wrong decision in plate recognition step at next section.

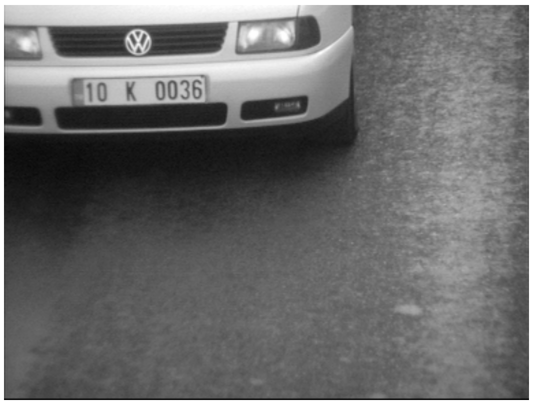


Figure 2.10 : A sample Input Image

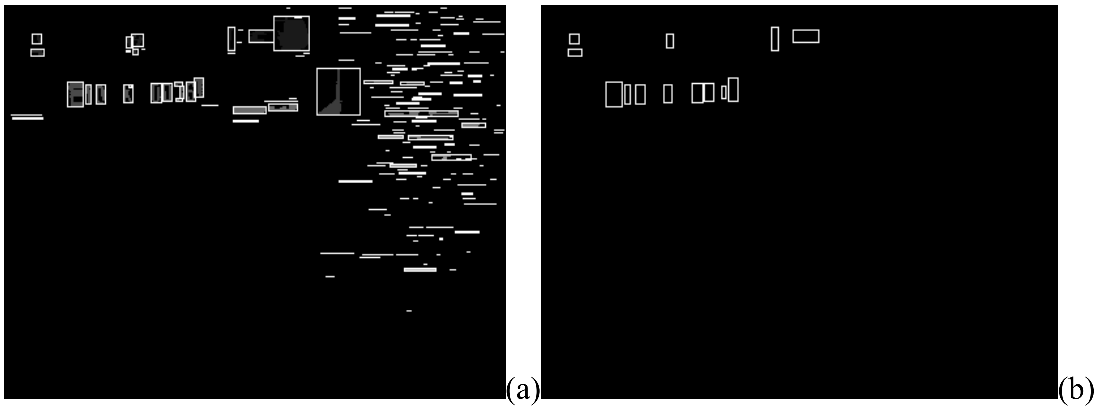


Figure 2.11 : Connected Component Stage (a) Blob surrounding rectangles, (b) reduced rectangles

2.5 Detecting Plate Candidates

The rectangles are in Figure 2.9 that belongs not only to plate characters but also some other fake ones. The plate location detection step similarly depends on a number of rules to group rectangles that belongs to a feature set. The idea is simple that all rectangles supplying similar features in terms of height and width may be a plate character candidate and the others are not. In addition to that, the plate consists of a number of characters. Thus an additional threshold will be used to at least 4 characters assumed to be included in a plate region. The width, height and number rules presented in [18] are used to find correct location of the plates in database. The algorithm is illustrated in Figure 2.12.

The width rule is based on the assumption that distance between two adjacent characters must be in a restricted window. The maximum distance may be predefined by considering result of the analysis of image attributes in selected database. As assuming that R1, R2 are plate characters and W is the function of width of characters, the merged region should satisfy the inequality equation below.

$$W(R_1 \cup R_2) < [W(R_1) + W(R_2) + G] \quad \textbf{(2.10)}$$

G is the maximum distance between two characters.

The height rule should satisfy the closeness of characters along vertical direction. On the other hand, the characters may be apparently tilt or different heights that may be the reason of vertical closeness decrease. The rule should satisfy this trade off.

Assuming that Y_1 and Y_2 are the lower board and the upper bound of height coefficient, and H is the function that gives height of any character. Y_1 and Y_2 should be as following equation (2.11) and (2.12)

$$Y_1 = \frac{maks(H(R_1), H(R_2))}{[H(R_1) + H(R_2)]} \quad \textbf{(2.11)}$$

$$Y_1 < Y_2 < 1 \quad \textbf{(2.12)}$$

The height rule is then as given equation (2.13).

$$[H(R_1) + H(R_2)]\ x\ Y_1 < H(R_1 \cup R_2) < [H(R_1) + H(R_2)]\ x\ Y_2 \quad \textbf{(2.13)}$$

The lower bound character number rule is based on the information that the total number of characters should be between four and eight. Only the lower bound is selected as a rule.

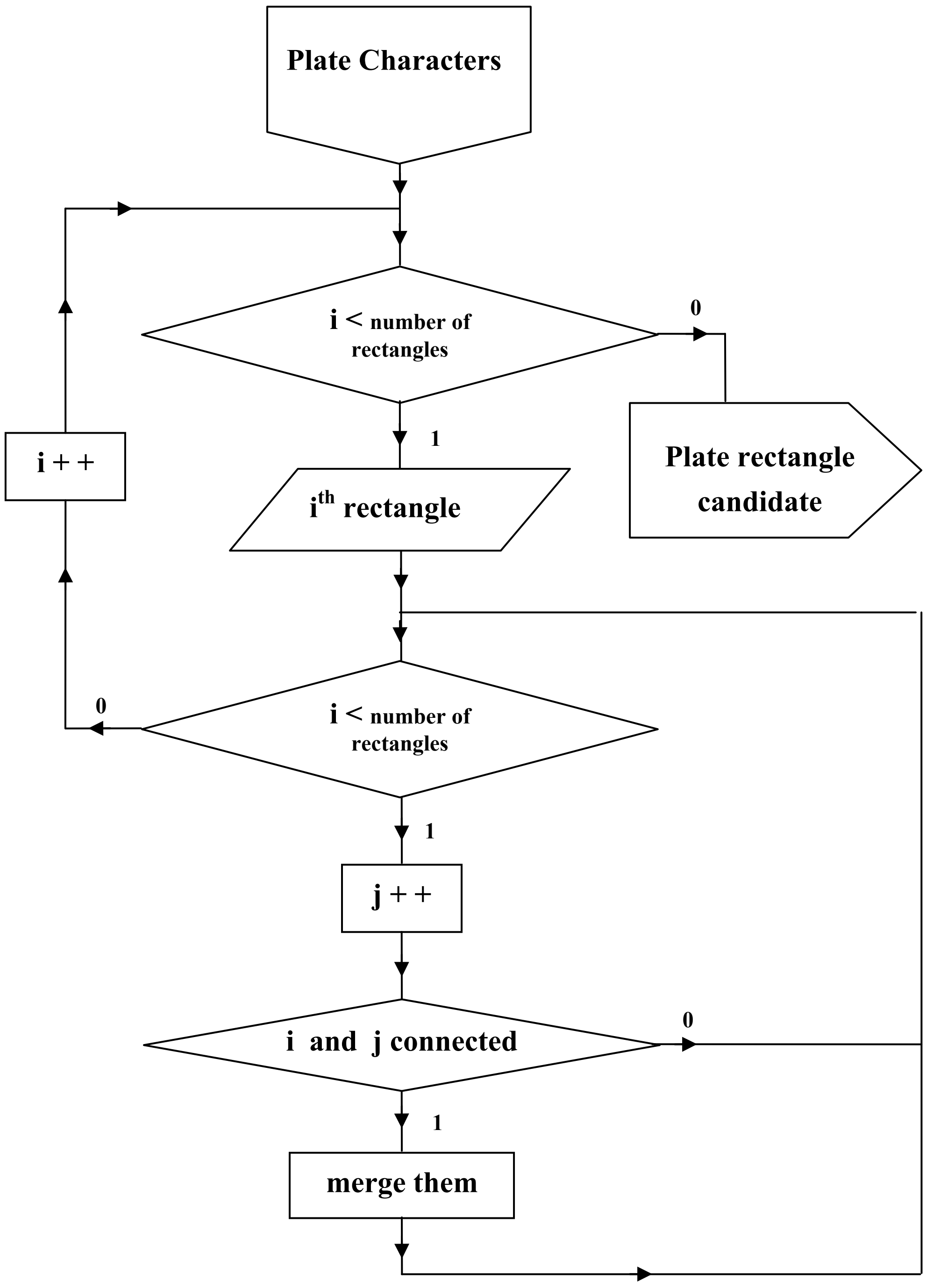


Figure 2.12 : Plate Candidates Detection Flow Chart

3. RESULTS

A number of experimental and system comparison like algorithm cycle times, process durations, memory requirements, and power consumption constraints between pc based systems and ADSP BF561 based embedded system is denoted in following sections in addition to given figures of the outputs captured at different stages of real time video processing during LP localization process.

3.1 Sample Results

These results belong to outputs of main algorithm modules spanning from digital image acquisition to display output.

CVBS output of video camera, in PAL standard, connected to video decoder gives a D1 signal. The corresponding video frame is interlaced signal in resolution of height equal to 243 and width equal to 720 pixels as given at Figure 3.1. Since the chroma data is not used in algorithm level, it is discarded.

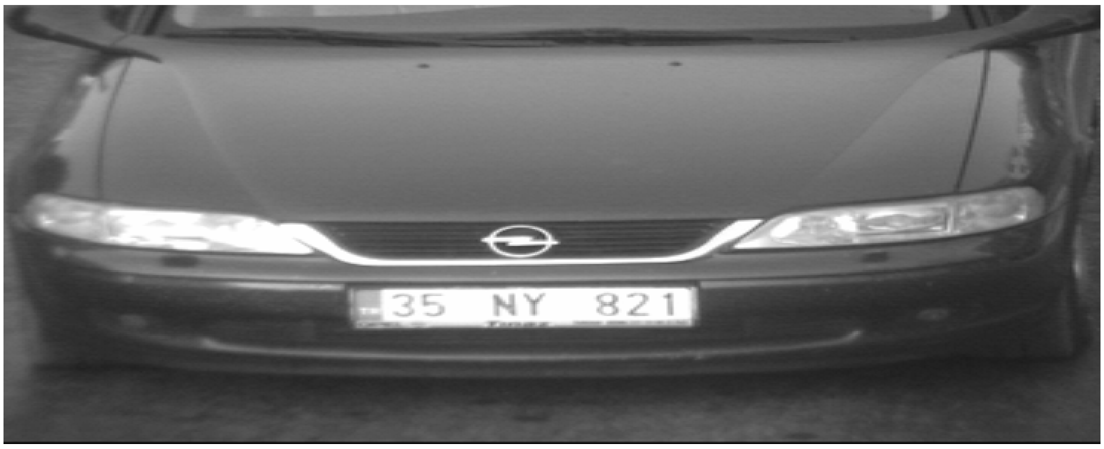


Figure 3.1 : Interlaced D1 Input Frame

Calculated gradient magnitudes of input image related only to horizontal direction in spatial domain is given at below Figure 3.2.

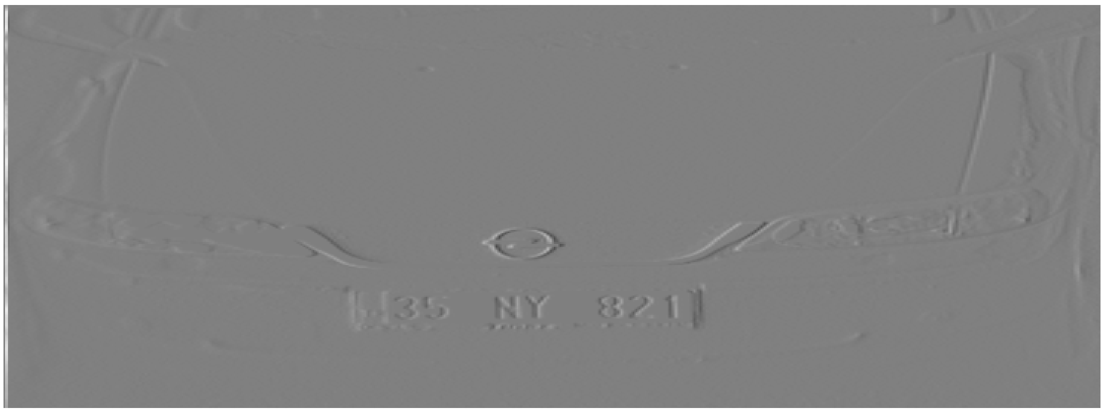


Figure 3.2 : Horizontal Gradients

The image applied local threshold with the adjacent negative and positive gradient companions and segmented with connected components analysis and surrounding indicator rectangles of regions are given at Figure 3.3 for sample experiment.

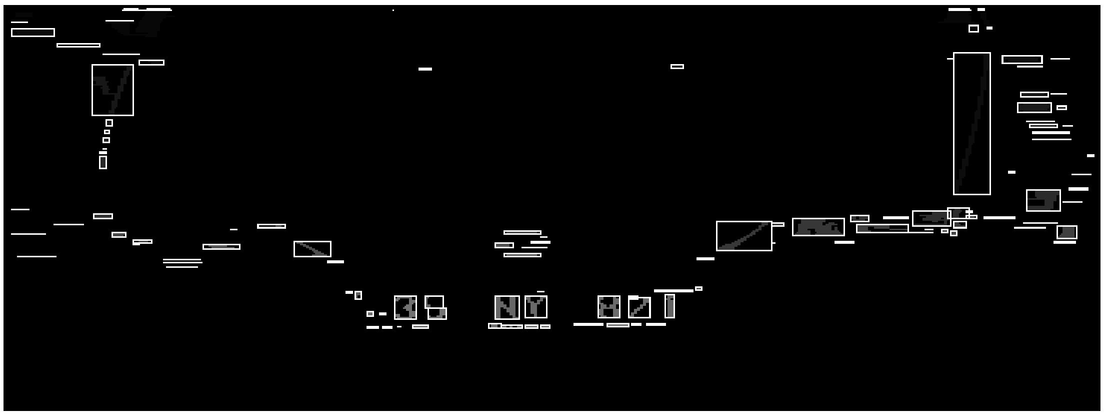


Figure 3.3 : Segmented Image

The process of merging of intersected rectangles, and plate character feature rules which are applied to the input in Figure 3.3 gives a reduction achievement as in Figure 3.4.

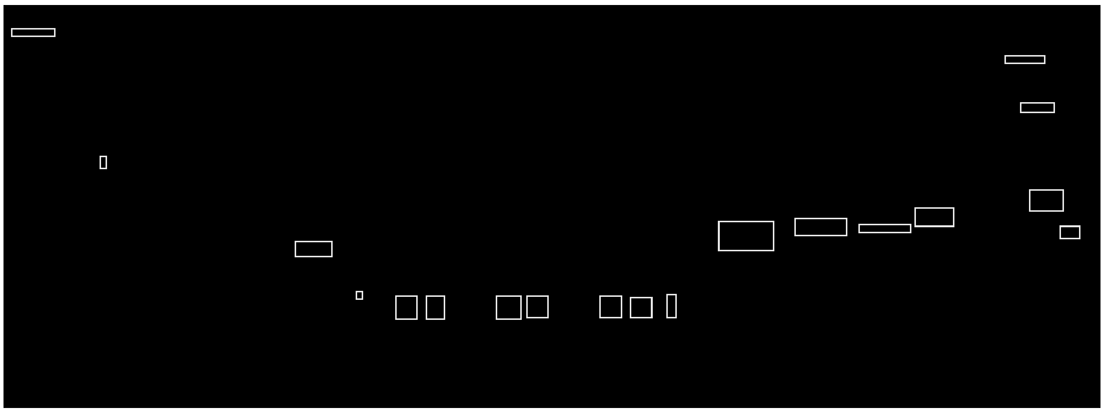


Figure 3.4 : Reduced Rectangles

The plate region restriction rules applied to output achieved at last section gives the correct location of the plate in given image at Figure 3.5.

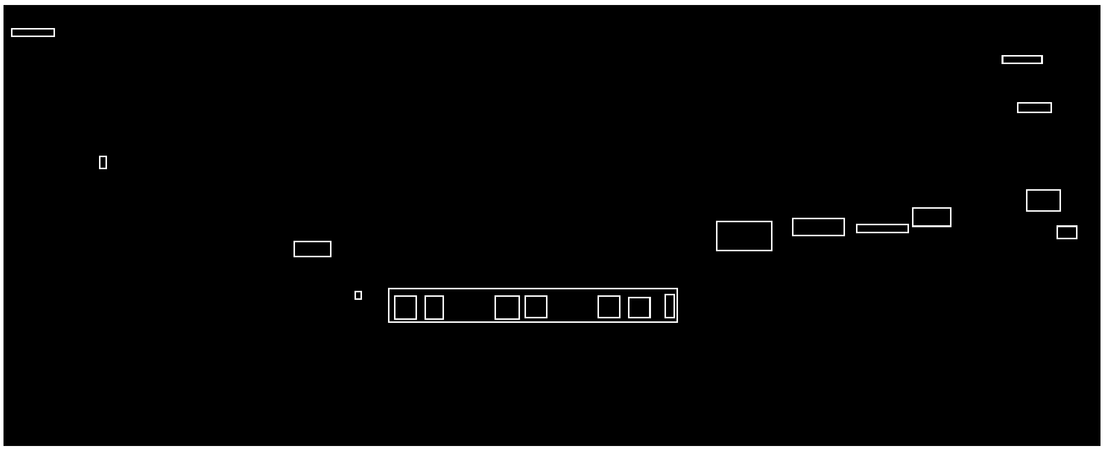


Figure 3.5 : Plate Localization

The localized plate region is now ready to be an input to recognition process. This may also be applied inside embedded platform or a server machine preferably as depending on system architecture and purpose.

Several responses to different input images focused on next to plate frame, surrounding region of plate, and along input spatial resolution is illustrated in Figure 3.6, Figure 3.7, and Figure 3.8 respectively.

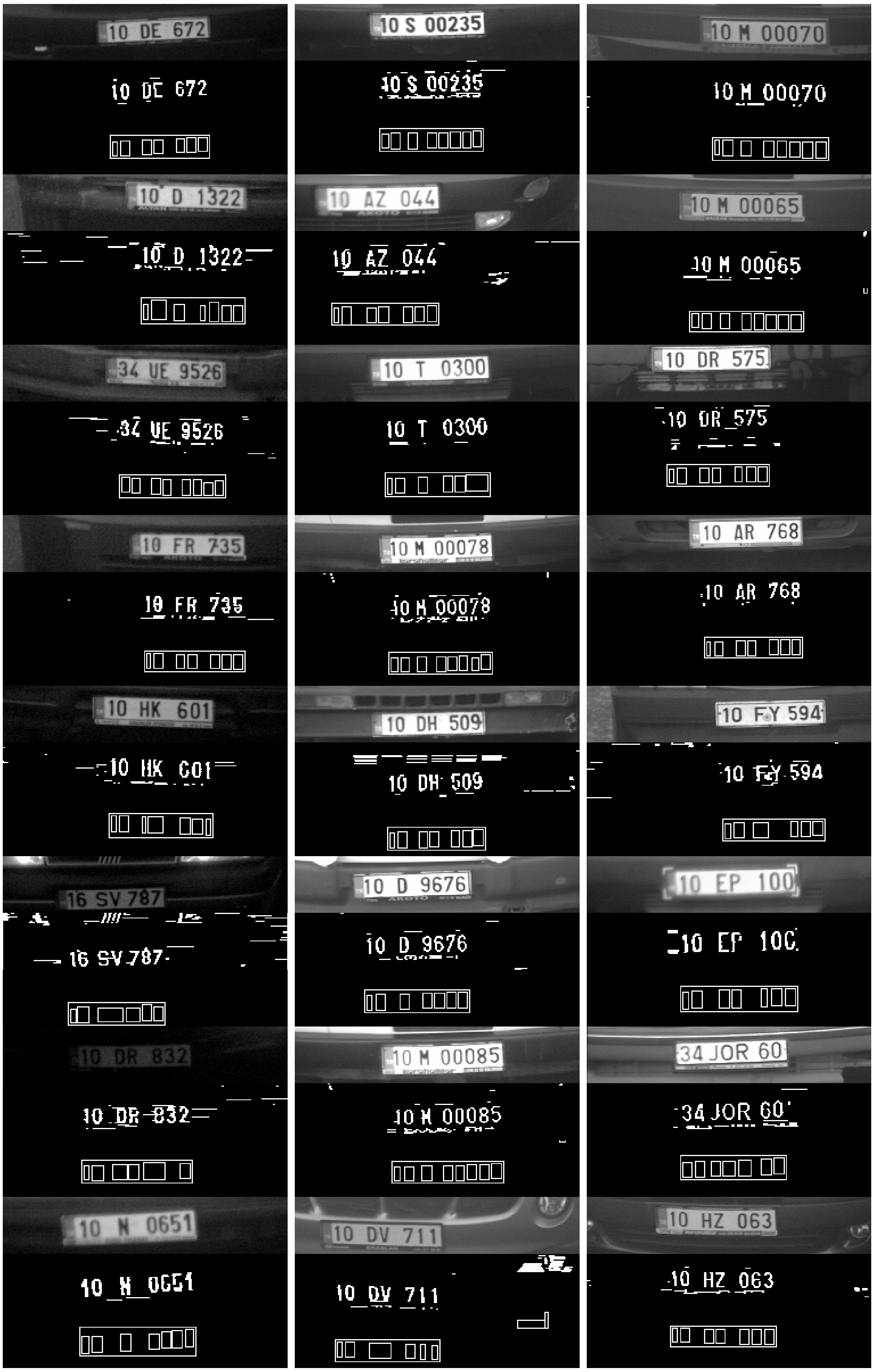


Figure 3.6 : Results focused on plate on plate

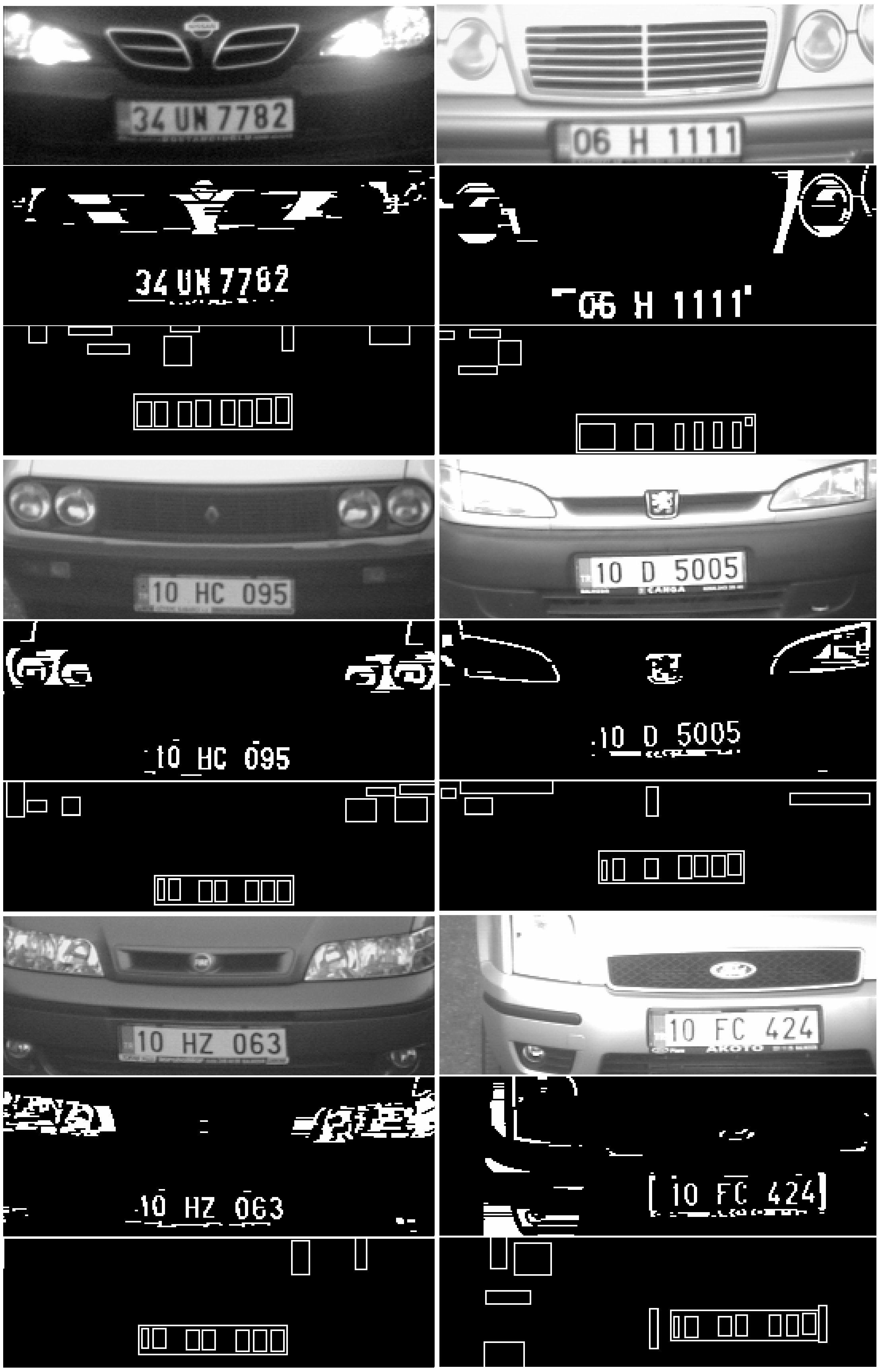


Figure 3.7 : Results subject to surrounding region of plate area

3.2 Experimental Results

Some of the results belonging to sample experiments during implementation of system design are given here in this section. There is no unique image or image sequence database to compare experimental results between different methods. Instead of localization results performance comparison in a predefined common database, the rates of correct and incorrect plate localizations is generally used a criterion in literature to measure the performance of algorithm as in Table 3.1

Table 3.1 : License Plate Localization Results

	True Positive	False Positive	False Negative
Rate	(1134/1238)	(97/1238)	(18/1238)
Results	%91.59	%08,07	%01,45

In case a plate candidate location is correctly matches to the target plate in a satisfactory range, it is marked as true positive. If a candidate is not found in case there is a valid target, this result is marked as false negative. False positive is used here to mention that a plate candidate is found in system but not matched to correct location. The similarity of candidates and target plates are calculated by using Tanimoto similarity [26]. This similarity criterion is simply determined as a proportion equal to the division of the intersection of target and candidate regions by combination of them.

There are totally 1238 images in vehicle license plate database which belongs to the similar camera placement parameters like height of camera location, distance to road, angle of view. It is required for detection rules of plate characters candidates and plate candidates to work unsurprisingly.

Some of the parameters used for correct detection of character candidates, and determine plate candidates are analyzed in detail in this section to figure how the range of various data effect the overall results.

As a result of trials with various levels for local threshold operation to get sufficient negative positive gradient pairs and obtain overall satisfactory results, it is shown in the Figure 3.9 that the assumption of the percentage of high order gradient values which is around %1-%2 of all gradients of input image works reasonably.

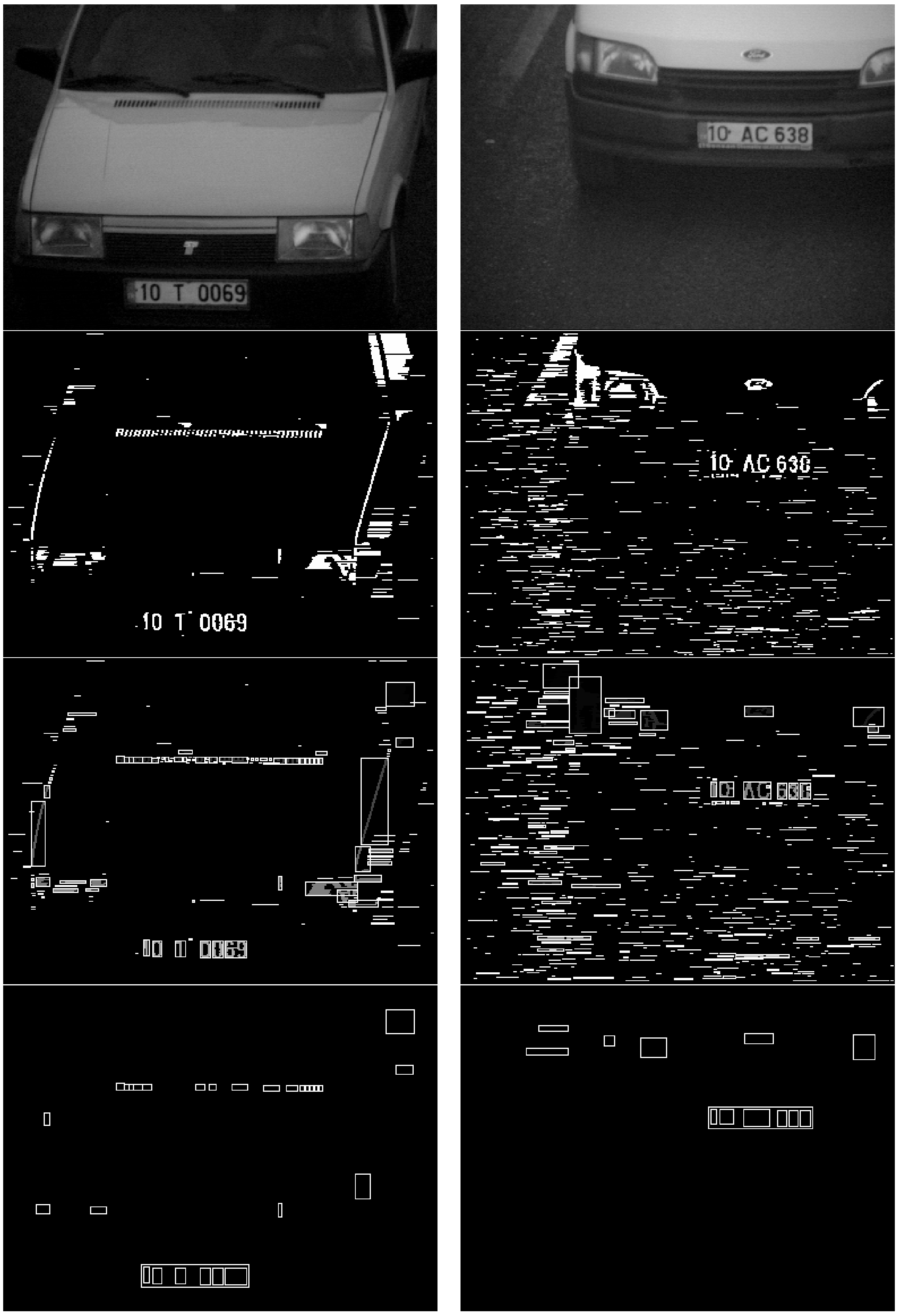


Figure 3.8 : Sample Results captured from video stream

The figure shows that the locations of plates are correctly determined for approximately %91 of total license plates by selecting the threshold level to satisfy high valued %2 percent of each gradients image. As the threshold level increases after this peek of true positives, true positive results start to decrease.

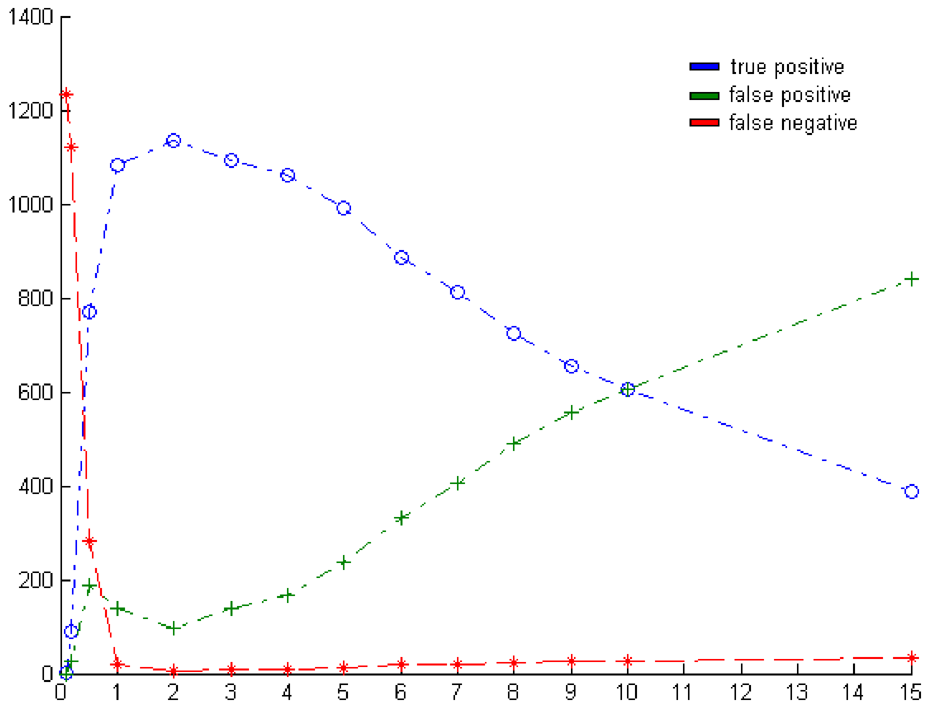


Figure 3.9 : Results with changing local threshold level

The stroke width which is one of distinction parameters for target character candidates to be separated from misleading candidates should be below a lower bound. This parameter is varied with different view and camera placement conditions. The lower bound of stroke width is analyzed for a predefined stroke rate determining the proportion of the segments of characters which length is below a stroke value over the total number of lines of character candidate. It gives the lower bound of stroke width within selected database.

The Figure 3.10 gives the lower bound of stroke length versus sample results.

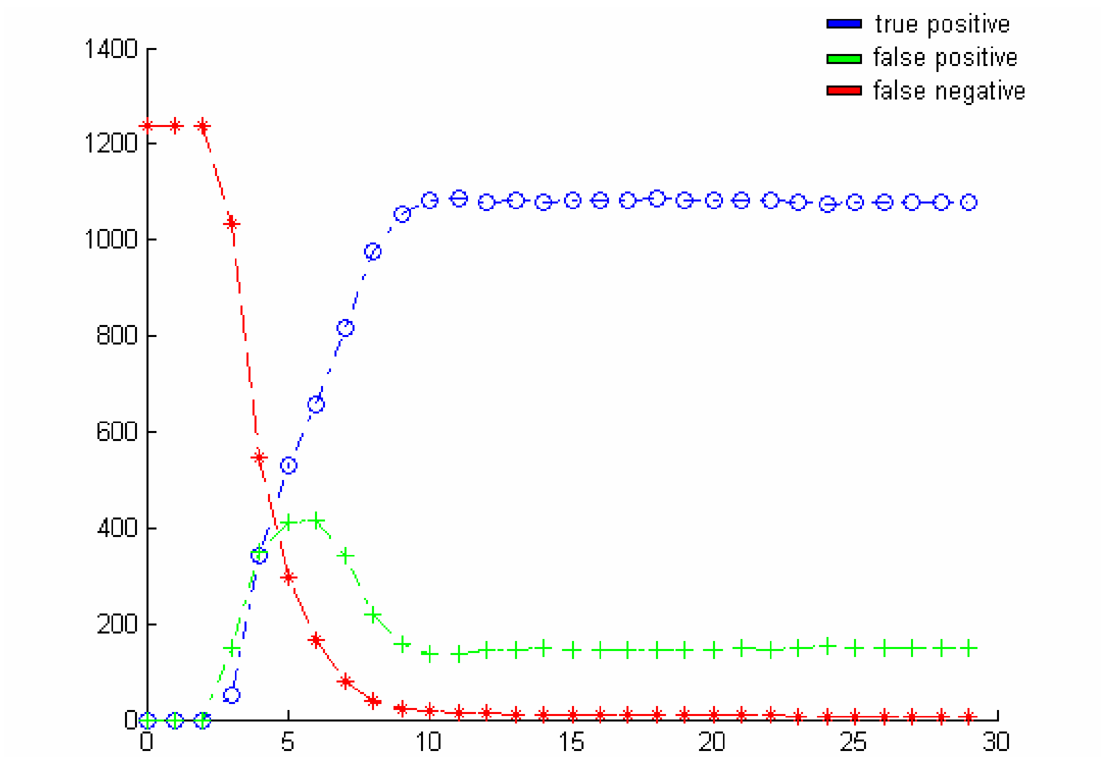


Figure 3.10 : Sample results versus character stroke length

While detecting license plate character candidates, the height of characters restricted with lower and upper bounds are one of utilization parameter. The corresponding minimum and maximum height boundaries analyzed in a range between 0 and 50 pixels are given in results at Figure 3.11 : .

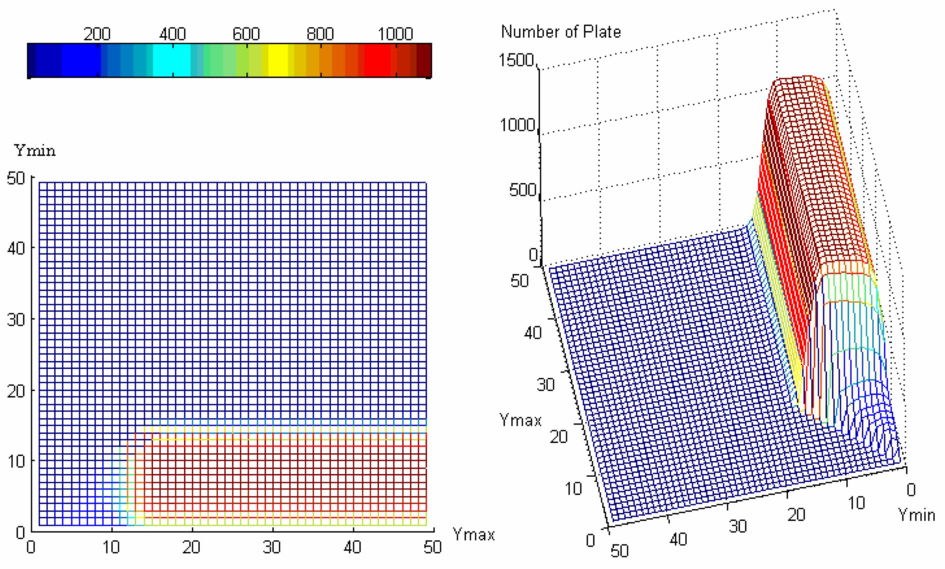


Figure 3.11 : Sample Results versus minimum and maximum character height

One of the important advantages aimed in this work is to satisfy the requirement of design criterions like compactness, low power and reduced memory requirements of system and algorithm process. Power consumptions are roughly compared within Table 3.2 for PC based architecture and embedded system.

Table 3.2 : Power Consumption Figure

	600 Mhz BF561 DSP	2.0 GHz Pentium PC
CONSUMPTION	600 – 800 mW	100 - 200 Watt

It is obviously visible that the memory requirement for a personal computer is very high as compared with an embedded design because of operating system and peripheral hardware requirements. It may be reviewed the memory requirement for both system in Table 3.3.

Table 3.3 : Memory Utilization

	Proposed Embedded System Requirements	PC Based System Requirements
Flash/HD Size	8 MB	256 MB
Ram Size	64 MB	1GB

Even though the selected DSP can not execute multiple instructions at the same time, it is possible to run them in parallel. Two data address generators provide addresses for simultaneous dual operant fetches from memory. The comparison table of the optimization applied to convolutions and filtering by utilizing capabilities of 64-bit, 32-bit and 16-bit instructions, hardware loops without CPU overloads, dual memory fetches used for parallel memory access and executions are given at Table 3.4.

It seems in Table 3.4 that CPU loads during assembly optimized convolution of an input image which spatial resolution is equal to 288x386 with a 5x5 filter mask is approximately 15 times faster than floating point non-optimized operation in terms of fix point architecture benefits. It should be mentioned that it is two times faster independently from fixed or floating point approach than any sequential process because of efficient computation of two output points for each of loop iteration.

Table 3.4 : Sample Functions Optimization Results

	Optimized		Non – Optimized	
	Cycles	Duration (sec)	Cycles	Duration (sec)
1D Conv per pixel	17,13	—	226	—
2D Conv per pixel	16,03	—	232	—
1D Conv of 288x384 Frame	9472639	0,015788	125457034	0,209095
2D Conv of 288x384 Frame	44340630	0,073901	643264167	1,072107

4. CONCLUSION

In this study, we proposed an embedded real-time implementation of a compact license plate recognition system in a suitable hardware. Since there is not a common license plate database, the results are analyzed due to the rates of correct and incorrect plate localization which is generally used as a criterion in literature to measure the performance of the algorithm. As a result, it seems that the presented algorithm is efficiently applicable method in an embedded DSP platform selected. The system design modules spanning from image acquisition controls and video output to localization application are implemented from scratch. DMA pipeline is programmed for efficient data transfers between different software modules like gradient image calculations, display memory and application memory movements. The design is restricted to low memory requirements successfully. The power consumption and compactness and portability features of embedded architecture are very attractive in spite of the complication and difficulty of system design during development, debugging and analysis of results as compared with high level PC development environments. Second core of the CPU may be added to software design for stage of recognition of the characters.

The license plate recognition system can be reviewed for several additional features like square plates, intractable lorry and bus images without changing the system architecture. Increasing system complexity should not be expected for these add-on feature sets which are suggested to be developed straightforward since corresponding additional software modules working on run length codes use less memory than the process like filtering or threshold process. These features enhance over all algorithm recognition results within an LPR system.

In future works, a wireless or wired link like ethernet or universal serial bus may be added to the system for remote control and corresponding outputs or numeric data of the output of recognition process to be transferred to operation control unit servicing high number of embedded processing units.

REFERENCES

[1] **Chen, H., Ren J., Tan, H., Wang, J.,** 2007. A Novel Method for License Plate Localization *Image and Graphics, Fourth International Conference,* Chengdu, China, 22-24 Aug. 2007, pp., 604-609.

[2] **Hsieh, J.W., Yu, S.H., Chen, Y.S.,** 2002. Morphology-Based License Plate Detection from Complex Scenes, *Proceedings of the 16th International Conference on Pattern Recognition (ICPR'02),* 11-15 August 2002, **(3)**, pp. 176-179

[3] **Hongliang B., and Changping, L.,** 2004 A hybrid license plate extraction method based on edge statistics and morphology. *17th International Conference On Pattern Recognition (ICPR'04),* (2), pp. 831-834.

[4] **Kamat, V., and Ganesan, S.,** 1995. An efficient implementation of the Hough transform for detection vehicle license plates using DSP'S. *Proceedings of the Real-Time Technology and Applications Symposium,* pp.58

[5] **Y. Yanamura, M. Goto, D. Nishiyama, M. Soga, H. Nakatani and H. Saji,** 2003. Extraction and tracking of the license plate using Hough transform and voted block matching, *IEEE IV2003 Intelligent Vehicles Symposium Proceedings,* Piscataway, USA, pp. 243-246.

[6] **Kahraman, F. ve Gokmen M.,** 2003. Gabor Süzgeçleri kullanılarak taşıt plakalarının yerinin saptanması, *Sinyal İşleme ve iletişim uygulamaları Kurultayı,* KÜ, Istanbul, 18-20 Haziran, 2003, s. 367-370.

[7] **S.Yohimori, Y. Mitsukura, M. Fukumi, N. Akamatsu and W. Pedryez,** 2004 *License plate detection system by using threshold function and improved template matching method,* Fuzzy Information Processing, NAFIPS '04, Alberta, Canada, 27-30 June 2004, pp. 357-362.

[8] **Shapiro, V., Gluhchev, G. and Dimov, D.,** 2006. Toward a multinational car license plate recognition system, *Machine Vision and Applications,* **17(3)**:173-183.

[9] **Dlagnekov L. and Belongie S.,** 2005. Recognizing cars, *Technical Report,* CS2005-0833, UCSD University of California, San Diego.

[10] **Kang J., Kang, M., Park, C., Kim, J., and Choi Y.,** 2004. Implementation of embedded system for vehicle tracking and license plates recognition using spatial relative distance, *In 26th International Conference on Information Technology Interfaces (ITI),* 1, **(167)**, pp. 172.

[11] **Bellas, N., Chai, S. M., Dwyer, M. and Cinzmeier D.,** 2006. *FPGA implementation of a license plate recognition soc using automatically generated streaming accelerators,* 20th International Parallel and Distributed Processing Symposium (IPDPS), Nice, France, pp. 8.

[12] **Arth, C., Limberger, F. and Bischof, H.,** 2007.Real-Time License Plate Recognition on an Embedded DSP-Platform, *Proceedings of the 3th Workshop on Embedded Computer Vision, IEEE International Conference on Computer Vision and Pattern Recognition,* 17-22 June 2007, Minneapolis, pp. 1-8.

[13] **Caner, H., Gecim, H.S., Alkar, A.Z.,** Efficient Embedded Neural Network Based License Plate Recognition System, *Vehicular Technology, IEEE Transactions on*: Accepted for future publication.

[14] **Arth, C.; Bischof, H.; Leistner, C.,** 2006. TRICam – An Embedded Platform for Remote Traffic Surveillance, *Computer Vision and Pattern Recognition Workshop,* 17-22 June 2006, pp. 125.

[15] **Alefs, B.,** 2006. Embedded Vehicle Detection by Boostin, *Intelligent Transportation Systems Conference, ITSC apos; 06.,* Toronto, Canada, 17-20 September 2006, pp. 536-541.

[16] **Bauer, D., Belbachir A.N., Donath, N., Gritsch, G., Kohn, B., Litzenberger M., Posch, C., Schön, P. and Schraml, S.,** 2007. Embedded Vehicle Speed Estimation System Using an Asynchronous Temporal Contrast Vision Sensor, Austrion Research Centers GmbH – ARC, Vienna 1220, Austrio, *EURASIP Journal on Embedded Systems* (Hindawi Publishing Corporation), pp. 12.

[17] Litsenberger, M., Kohn, B., Gritsch, G., Donath, N., Posch, C., Belbachir, A., N., Garn, H., 2007. Vehicle Counting with an Embedded Traffic Data System using an Optical Transient Sensor, *Proceedings of the 2007 IEEE Intelligent Transportation Systems Conference,* Seattle, USA, Sept. 30 – Oct. 3, 2007.

[18] Beratoğlu M., S., 2003. Araç Plaka Yerinin Saptanması, *Yüksek Lisans Tezi,* İ.T.Ü-Fen Bilimleri Enstitüsü, İstanbul.

[19] ADSP-BF561, 2007. Hardware Manual, Analog Devices, USA.

[20] Çapar, A., Beratoğlu, M. S., Taşdemir, K., Kılıç, Ö., Gökmen, M., 2003. İTÜ Araç Plaka Tanıma Sistemi, *11. Sinyal İşleme ve İletişim Uygulamaları Kurultayı (SİU)* 2003 İstanbul, pp.371-374.

[21] Canny, J. F., 1986. A computational approach to edge detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* **8**, 679-698.

[22] Katz D. J., Gentile R., 2005. Embedded Media Processing, Newnes, Elsevier, pp. 432.

[23] Cui, Y. and Huang, Q., 1998. Extracting characters of license plates from video sequences, *Machine Vision and Applications,* **10(5/6)**, 308-320.

[24] Trein, J., Schwarzbacher A. Th., Hoppe B., Noffz K. H. and Trenschel, T., 2007. Development of a FPGA Based Real-Time Blob Analysis Circuit, *ISSC,* Derry, Northern Ireland, 13-14 Sept 2007.

[25] Rachakonda, R. V., 1995. High-Speed Region Detection and Labeling Using an FPGA-based Custom Computing Platform, *in W. Moore, W. Luk, Eds., Lecture Notes in Computer Science 975 – Field Programmable Logic and Applications,* pp. 86-93, London: Springer.

[26] Duda, R. O. and Hart,PE , 1973. Pattern Classification and Scene Analysis, wiley-interscience publication, NY.

CIRCULUM VITAE

Özgür Bulkan, 1978 yılında İstanbul'da dünyaya geldi. İlk öğrenimi Faik Reşit Unat İlköğretim Okulu'nda (1985/1990), orta öğrenimini Göztepe Orta Okulu'nda (1990/1993), ve liseyi Haydarpaşa Lisesi'nde (1993/1996) tamamladı. İstanbul Üniversitesi Elektrik Elektronik Mühendisliği Bölümünden 2000 yılında mezun oldu. 2000 yılından bugüne sırası ile Profilo Telra A.Ş., Digiturk ve STMicroelectronics şirketlerinde sayısal görüntü teknolojileri alanında araştırma, geliştirme ve uygulamaları ile ilgili çalışmalarda bulundu.