4th International Conference on Automation, Control Engineering and Computer Science (ACECS - 2017)
Proceedings of Engineering and Technology – PET
Vol.19, pp.36-40

Hardware Implementation of Real-Time Image Segmentation Algorithms using TMS320C6713 DSP and VM3224K2 Daughter Kit

Boucif BEDDAD[#1], Kaddour HACHEMI[*2]

[#*]*LTC Laboratory, Faculty of Technology, Dr. TAHAR Moulay University*
Saida, Algeria

[1]boucif.beddad@univ-saida.dz
[2]kaddour.hachemi@univ-saida.dz

Abstract— **Digital image processing continues to enable the multimedia technology revolution that we are experiencing today. The main purpose of this project is to create a code for implementing license plate segmentation from vehicle image in various luminance conditions. The basic process involves is to take image as input from CCD camera which is to be segmented in DSP board using a very efficient segmentation algorithm based on the analysis of vertical and horizontal histogram with the edge processing methods. In this paper we have also used DSP Simulink Blocksets in MATLAB for simulation purposes and C6713 DSK, Code Composer Studio v3.1 for code developing and implementation. The segmentation technique and proposed algorithm were well implemented practically in real-time by using the floating-point processor TMS320C6713 DSP chip manufactured by Texas Instruments. VM3224K2 Daughter Kit (LCD) is embedded with DSK6713 Kit to display the output image also L2 cache memory and the compiler optimization techniques were tuned for best performance improvement.**

***Keywords*— TMS320C6713, VM3224K2 Kit, Code Composer Studio, Segmentation, Edge Detection, Thresholding Techniques**

I. Introduction

Image processing is a method to perform some operations on a digital image in order to get an enhanced image or to extract hidden information that can be interpreted and used to several purposes. Image segmentation has a promising future as the universal segmentation algorithm and has become the focus of contemporary research; it's an integral step in any image processing that some applications requires a simple division of image into homogeneous regions while others require more accurate detection, so for this reason we need a special unit like a DSP processor that we use for digital signal and image processing in real time and high-speed data transfer.

Digital signal processor is one of the core technologies in rapidly growing application that become a key component in many domains. A C6x processor can be used as a standard general-purpose digital signal processor programmed for a specific application. DSP is generally more expensive since it has more real estate. Several factors, such as cost, power consumption, and speed, come into play when choosing a DSP. The Specific-purpose digital signal processors are available with different feature, such as faster execution speed.

Digital signal processors such as the TMS320C6x family of Texas Instruments are like fast special purpose microprocessors with a specialized type of architecture and an instruction set appropriate for signal processing. All DSP techniques have been very successful because of the development of low-cost software and hardware support. TMS320C6x DSP is considered to be TI's most powerful processor for applications requiring intensive computations.

II. Digital Image Processing

An image defined as a two-dimensional function f(x, y); x and y are spatial coordinates. The amplitude of the 2D function f at any pair of coordinates (x, y) is called intensity or gray level of the image at that point. When f, x and y are all finite and discrete quantities, these elements are referred to as pixels. These amplitudes composed as a vector of data where the Color image has three components RGB at each point. License plate recognition is a type of technology, mainly software that enables computer systems to read automatically the registration number of vehicles from digital RGB images.

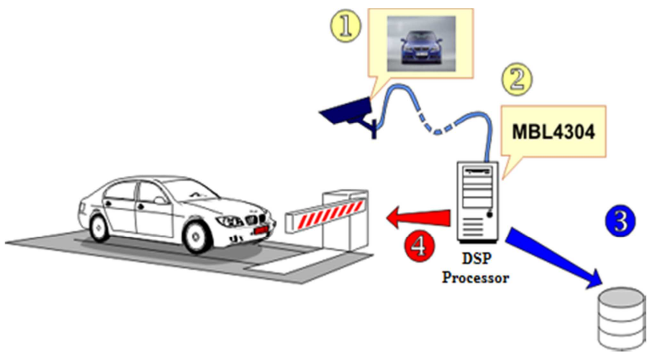

Fig. 1 Automatic Number Plate Recognition System (ANPR)

RGB image is a 3 dimensional combination of RED, BLUE and GREEN colors. In our work we have represented the input image using Y-Cb-Cr format then we displayed by converting into RGB format using the mathematical equations:

$$R = Y + 1.402\,(Cr-128) \quad (1)$$
$$G = Y - 0.34414\,(Cb-128) - 0.71414\,(Cr-128) \quad (2)$$
$$B = Y + 1.772\,(Cb-128) \quad (3)$$

ISSN 2356-5608

A. RGB to Grayscale Image Conversion

The grayscale image is obtained from the RGB image by combining 30% of RED, 60% of GREEN and 11% of BLUE. Then 2D matrices are stored in MATLAB software and displayed as gray scale image. The intensity of this grayscale image is ranging from 0 to 255 where 0 resembles purely black color and 255 resemble purely white color. Any intermediate values between this rang vary in intensity from black to white. The following equations are used to convert the image from RGB to YUV:

$$Y = 0.299R + 0.587G + 0.114B \quad (4)$$
$$U = -0.147R - 0.289G + 0.436B \quad (5)$$
$$V = 0.615R - 0.515G - 0.100B \quad (6)$$

B. Image Histogram

Histogram of an image is a graphical representation of the number of pixels at each different intensity value found in that image. The x axis shows the gray level intensities and y axis shows the frequency of these intensities. There are 256 differents possible intensities; the histogram will graphically display 256 numbers showing the distribution of pixels values.

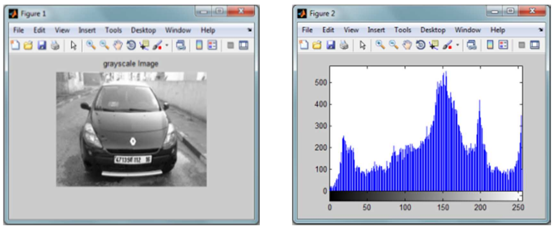

Fig. 2 Grayscale Image Histogram

C. Image Segmentation Methods

Segmentation is the division of an image into regions or categories which correspond to different objects or parts of objects. Every pixel in an image is allocated to one of a number of these categories. It is an essential step in image processing and also it is a critical stage of image analysis. To define the segmentation: supposing the image is represented by W, Let W denotes the whole image region R_i and i=1, 2,…, k. are disjoint nonempty regions of W, consists of the following conditions:

$$\bigcup_{i=1}^{k} R_i = W$$

$$\text{For All} \begin{cases} \text{i, j and } i \neq j, \text{there exits } R_i \cap R_j = \emptyset \\ i = 1,2,..,k, \text{it must have } P(R_i) = \text{True} \\ i \neq j, \text{there exits } P(R_i \cup R_j) = \text{False} \\ i = 1,2,\ldots,k, R_i \text{ is connected component} \end{cases}$$

Many methods have been proposed most of which are based on histogram analysis, clustering and thresholding steps. The following method are used in the new proposed algorithm:

1) *Histogram Thresholding Methods:* The Thresholding is one of the first low-level image processing techniques that he used to extract an object from its background by assigning an intensity value for each pixel. This threshold value T_r is determined by applying differences operation between the pixel intensity values. In general, we have ρ(x, y) represent the Local property of pixel and (x, y):

$$G(x, y) = \begin{cases} 0 & \text{if } f(x,y) \leq T_r \\ 1 & \text{if } f(x,y) > T_r \end{cases} \quad (7)$$

Where:

$$T_r = T[(x,y), \rho(x,y), f(x,y)] \quad (8)$$

Global Thresholding T_{rg} is calculated for all pixels which utilize the image gray level histogram. It depends just on a neighborhood of the function f(x,y):

$$T_{rg} = T[f(x,y)] \quad (9)$$

Local Thresholding T_{rL} depends not only on its gray level but also its local information that is to say, the gray levels of neighboring pixels, so we define the local threshold as:

$$T_{rL} = T[\rho(x,y), f(x,y)] \quad (10)$$

Dynamic Thresholding T_{rd} is an image's histogram which defines the gray level distribution of its pixels, he depends on the position coordinates (x,y) of f(x,y):

$$T_{rd} = T[(x,y)] \quad (11)$$

2) *Sobel Edge Detection:* It use a pair of 3x3 convolution masks, one estimating gradient in the x-direction and the other estimating the gradient in y–direction. The components of the gradient may be found using the following approximation:

$$\frac{\partial f(x,y)}{\partial x} = \gamma_x = \frac{f(x+dx,y) - f(x,y)}{dx} \quad (12)$$
$$\frac{\partial f(x,y)}{\partial y} = \gamma_y = \frac{f(x,y+dy) - f(x,y)}{dy} \quad (13)$$

Where dx and dy measure distance along the x and y directions respectively. In discrete images, one can consider dx and dy in the terms of numbers of pixel between two points. dx = dy = 1 (pixel spacing) is the point at which pixel coordinates are (i, j):

$$\gamma_x = f(i+1, j) - f(i,j) \quad (14)$$
$$\gamma_y = f(i, j+1) - f(i,j) \quad (15)$$

The magnitude Mn of the vector Δf is denoted as:

$$\gamma = Mn(\Delta f) = [\gamma_x^2 + \gamma_y^2]^{1/2} \quad (16)$$

The absolute gradient magnitude is then given and often approximated with [16], [17]:

$$|\gamma| = |\gamma_x| + |\gamma_y| \quad (17)$$

Where γ_x is for x direction and γ_y is for y direction. Finally the gradient direction ϑ is given by:

$$\vartheta = \tan^{-1}\left[\frac{\gamma_y}{\gamma_x}\right] \quad (18)$$

ISSN 2356-5608

III. HARDWARE AND SOFTWARE TOOLS

Now, we will present the hardware & software tools that are used in our research that are: TMS320C6713DSK toolkit for the processing, Code Composer Studio incorporates a C compiler, an assembler, and a linker to generate C6x executable files, VM3224K2 daughter card is also compatible with Image processing algorithms, as we are able to display only input and output. Rest of the processing is done in C6713.

IV. PROPOSED METHODOLOGY

This paper mainly focuses on the development of a new algorithm for extraction the number plate from live vehicle image. Proposed method has composed of the following steps:

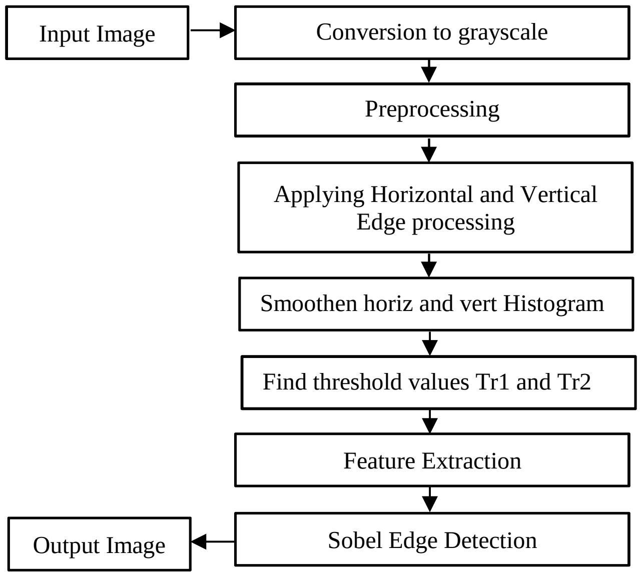

Fig. 3 Flowchart of the Proposed Algorithm

In the first step, pre-processing is applied by using a median filter to convert the input image into an enhanced gray scale image. In the second step the image is processed using row-wise and column-wise to find a common region having maximum horizontal and vertical histogram value also the column-wise and row-wise intensity differences are calculated. The minimum difference Min(r) and maximum difference values Max(r) are determined from the row vector and Min(c), Max(c) are determined from the column vector. The common region is considered as highest probability of containing a license plate number; it's determined by using histogram matching [1] technique. Further the resultant image is smoothened by applying a smoothing filter [7] to eliminate the distinctly marked pixels. Feature extraction of interested regions is performed through histogram based horizontal and vertical edge processing using a fixed threshold technique and also a fixed range method that respect to both threshold values 'Tr1' and 'Tr2' of the horizontal and vertical histograms :

$$\text{i.e.}, \text{Tr1} = \frac{\text{Min(row)}+\text{Min(col)}}{2} \quad (19)$$

$$\text{i.e.}, \text{Tr2} = \frac{\text{Max(row)}+\text{Max(col)}}{2} \quad (20)$$

V. IMPLEMENTATION

Once proposed algorithm was developed, it was completely verified in Matlab with multiple input images. Then the in-built functions of MATLAB were replaced by user-defined functions in Matlab Simulink Model that we uses different Blocksets of "Video and Image Processing Blockset" library, here we present different blocks used for this implementation.

TABLE I
DIFFERENTS SIMULINK BLOCKSET USED

N°	Matlab Simulink Blocksets		
	Blocks	**Library**	**Quantity**
1	From Video Device	L1	01
2	Color space conversion	L2	02
3	Median Filter	L3	01
4	Video Viewer	L4	03
5	Embedded MATLAB Function	L5	01
6	Image data type conversion	L2	02
7	Edge Detection	L3	01

L1: Video and Image Processing Blockset> Sources
L2: Video and Image Processing Blockset> Conversions
L3: Video and Image Processing Blockset> Analysis & Enhancement.
L4: Video and Image Processing Blockset> Sinks
L5: Simulink> User-Defined Functions

The proposed simulink model is given as follows:

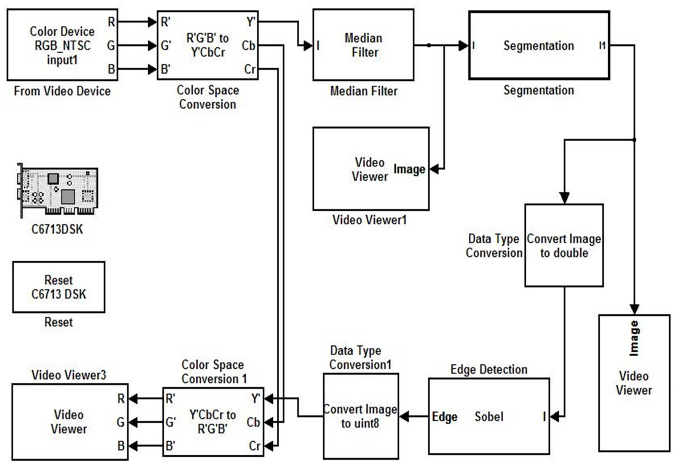

Fig. 4 Flowchart of the Proposed Simulink Model

On Matlab Simulink, there is a library containing a specific blocks for each type of DSP card, from this library we add the C6713 block then we selects processor type and also clock frequency value. Concerning the passage from DSP Simulink Model to the CCS, MATLAB has developed a tool "Link for code composer studio" which allowed him to communicate with the Code Composer Studio for compilation, real time debugging, and linking program in Integrated Development

ISSN 2356-5608

Environment (IDE). So with using this tool, the proposed Simulink model is translated into CCS project written in C language that could be compiled and run on C6713. Fig 11 shown the different steps of the passage from simulink to CCS.

```
### Loading TLC function libraries

.....
### Initial pass through model to cache user defined code
.
### Caching model source code
.........................
### Writing header file Segmentation_Sobel_Edge_Detection_types.h
### Writing header file Segmentation_Sobel_Edge_Detection.h
.
### Writing header file Segmentation_Sobel_Edge_Detection_private.h
### Writing source file Segmentation_Sobel_Edge_Detection.c
### Writing source file Segmentation_Sobel_Edge_Detection_data.c
.
### Writing source file Segmentation_Sobel_Edge_Detection_main.c
### TLC code generation complete.
### Creating project in Code Composer Studio(tm)...
```

Fig. 5 Flowchart of CCS Project Creating from Matlab Simulink

VI. EXPERIMENTAL RESULT

This section presents the results obtained after successful execution in CCS v3.1 and implementation in C6713 DSK.

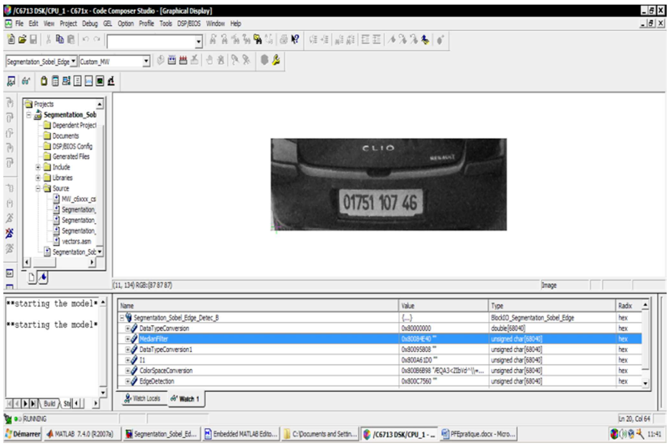

Fig. 6 Input Image Converted to Grayscale

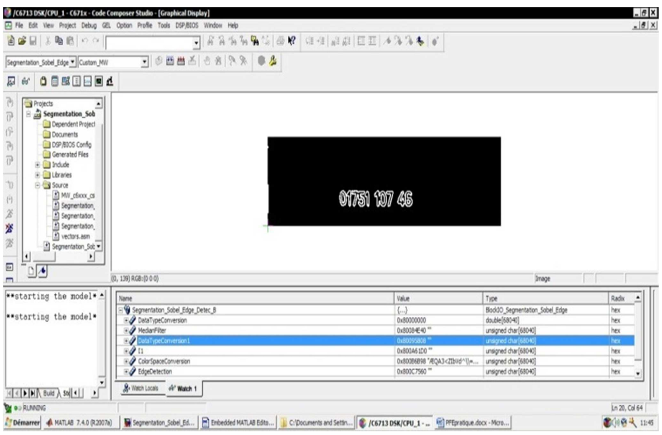

Fig. 7 Output Image Converted to RGB565

VII. OPTIMIZATION

In order to improve the total execution time on the chip, various optimization techniques were used to check the performance: C code, CCS compiler and L2 cache memory. The number of clock cycles required for each function is calculated and compared by a method called profiling for benchmark. The improved performance after adding these optimization techniques is shown in the table II:

TABLE III
PERFORMANCE IMPROVEMENT ACHIEVED

N°	Success Using Our Optimization Techniques		
	Function	**No. of CPU Cycles (Without Optimization)**	**No. of CPU Cycles (With Optimization)**
1	R'G'B' to Y'CbCr	3924756	2961921
2	Median Filter	91295273	67974354
3	Segmentation	199877799	148843146
4	Convert Image to double	3402892	2537678
5	Sobel Edge Detection	46642594	34720815
6	Convert Image to uint8	3635678	2721520
7	Y'CbCr to R'G'B'	3924789	2961847

VIII. CONCLUSION AND FUTURE WORK

In this paper, the hardware implementation of image segmentation algorithm is done. Also the performance of this proposed algorithm has been tested on several vehicle plates and provides very satisfactory results. Like we can see this result gives the conclusion that simulink based techniques are easy to understand and implement on DSP. This project can be used further as a base to develop systems for recognition which can be interfaced with database containing information.

ACKNOWLEDGMENT

I take this opportunity to express our sincere gratitude to all my staff members of LTC for their help and constant support.

REFERENCES

[1] G. Sudhakar Babu, K. Chandra Sekhara Reddy, "Automatic License Plate Recognition System," IJERA. Image Process, Vol. 3, pp.870-876, Jul-Aug 2013.

[2] N. Shobha Rani, O. P. Neethu, N. Ponnath, "Automatic Vehicle Tracking System Based on Fixed Thresholding and Histogram Based Edge Processing," IJECE. Vol. 5, pp. 869-878, August 2015.

[3] B. Sharmila, N. Karalan, "Image Processing on DSP Environment Using OpenCV," IJARCSSE, Vol 5, pp. 489~493, February 2015.

[4] U. Nemade, K.Shah, "Real Time Speech Recognition Using DSK TMS320C6713", IJARCSSE, Vol 4, pp. 461-469, January 2014.

[5] D. Nagpal, L. bhambhu, "Implementation of Simulink Based Model Using Sobel Edge Detector for Dental Problems," IJECSE. Image Process, Vol. 3, pp. 254-262, October 2013.

[6] P.Daniel Ratna Raju, G.Neelima, "Image Segmentation by using Histogram Thresholding," IJCSET, Vol 2, pp. 776-779, January 2012.

[7] O. R. Vincent, O. Folorunso, "A Descriptive Algorithm for Sobel Image Edge Detection," InSITE, Vol. 10, pp. 097-107, May 2009.

[8] J. M. Elham, J. M. Ahmed, "Design Study Sobel Edge Detection", IJAIEM, Vol. 2, Issue 12, pp. 248-253, December 2013.

ISSN 2356-5608