A FPGA-Based Method for License Plate Localization

Y.H. Tan, Y. Xin, X.J. Zhai
College of Electrical & Information Engineering
Hunan University
Changsha, China

***Abstract*—As a significant stage in an Automatic Number Plate recognition (ANPR) system, License plate localization (LPL) is often computationally expensive. A novel LPL algorithm using morphological operations is proposed in the paper. The algorithm has the property of low complexity and high detection rate. The algorithm has been proved to successfully implement and using the Mentor Graphics RC240 FPGA board equipped with a 4M Gates Xilinx Virtex-4 LX40 and a database of 2000 image. Results show that the proposed method achieves 98.4% detection rate within 2.7ms at the expense of only 2% of the available area in the FPGA.**

Keywords- license plate localization; FPGA; automatic number plate recognition

I. INTRODUCTION

An automatic Number Plate Recognition (ANPR) system has the function of tracking, identifying and monitoring moving vehicles. These systems are rapidly becoming applied for a vast number of applications such as automatic congestion charge systems, access control, tracing of stolen cars, or identification of dangerous drivers [1]. The steps involved in an ANPR system are image capture, image procession and plate recognition. In the image procession phase, two tasks are included, i.e. plate localization and character recognition. So it is often computationally intensive. Plate recognition is to separate License Plate (LP) area from Non-License Plate (Non-LP) area, and then plate adjustment is continued.

Current LP methods can mainly be classified into three classes: edge-based algorithms [2-5], color-based algorithms and texture-based algorithms[6-8]. The image processing algorithms requires strong computing power, so it is often required to choose high performance workstations and expensive supercomputers. However, the cost, compactness and power issues that come with these solutions motivate the search for other platforms. Recent improvements in the computing power of Programmable Gate Arrays (FPGAs) and Digital Signal Processors (DSPs) have motivated researchers to consider them as low cost solution for accelerating such computationally intensive task [2]-[4]. Thus, high performance computer workstations are necessary. An alternative solution is to choose FPGAs and DSPs [2]. These devices can be used as low-cost System-on-chip solution that allows the processing FPGA or DSP-based unit to be placed within an ANPR cameras housing to create intelligent cameras #C namely cameras that record and process images for sending back to a server. A variety of algorithms used in these systems includes AdaBoost [2][9], (SVM) [9], Gabor filter [10], morphological operation [11] and background modeling [2] and pixels classification [12]. The used algorithms are either computationally expensive or they have a low detection rate. The only work that involves the use of FPGA implement is the work in [12]. Results show that it has the fastest processing speed to locate LP with a relatively low detection rate compare to other existing work.

This paper presents a speed and area-efficient architecture based on a new low complexity LP localization method suitable for FPGA implementation. This method adopts open and close morphological operations [4]. By optimizing the ANPR algorithms to take advantage of technical features and innovations available within new FPGAs, such as reconfigure ability, development time, and rapid prototyping, it will be possible to replace the 3GHz roadside computers with small in-camera dedicated platforms.

II. THE LICENSE PLATE LOCALIZATION ALGORITHM

The feature used to locate the plate must be robust to the changes of lighting conditions and view orientations. Morphological open (MO) and morphological close (MC) operations are used to extract the contrast features within the plate. The proposed new algorithm is shown in Figure.1, consists of three main stages [4], i.e., MO and MC for extracting features, selection of candidate regions and validation.

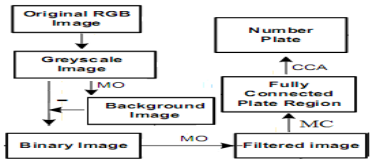
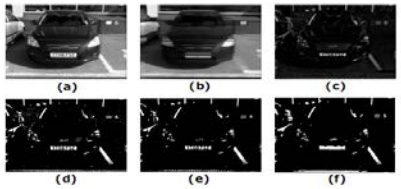


FIGURE I. BLOCK DIAGRAM OF THE PROPOSED LPL ALGORITHM.

In the feature extraction stage, morphological operations are adopted to enhance the plate region, and simultaneous minimize the pixels of non-plate region. We first obtain background image of the original RGB plate image using MO operations. And then we calculate the abstraction of the greyscale image and background image to highlight the plate region. In the following stage, the image is changed to binary image and another MO operation is used to filter out some pixel groups that smaller than the Structure Element (SE) of open operation. Finally, a MC is used for connecting the pixels of plate region. All stages have been implemented and tested and Figure. 2 shows the result of each stage.



(a) Original greyscale image; (b) Background image; (c) Enhanced plate region; (d) Binary image; (e) Filtered image; (f) Fully connected plate region.

FIGURE II. THE RESULTS OF EACH STAGE FROM THE PLATE FEATURE EXTRACTION PART.

The next step focuses on scanning image pixels on the basis of Connected Component Analysis (CCA) and labels them to identify connected pixel regions apply. And then the plate region is extracted from the original greyscale image, which is based on the coordinates of the identified rectangular region. Finally, this identified region will be validated by the character segmentation. That is, the feature segmentation will be passed, and otherwise, the next candidates will be chose for the following program.

III. PROPOSED LICENSE PLATE LOCALIZATION ARCHITECTURE

The proposed LP localization architecture consists of three major modules: a memory reader and converter, a morphological operations module and CCA module.

The first module is the memory reader and converter. The memory reader part reads the RGB values for each pixel from the original RGB image and assign a position coordinate to it. The converter is used for the standard RGB (24 bits) image to greyscale (8 bits) conversion.

The second unit is the operation module, which consists of two sub-modules, MO and MC operations. The MO operation here is an erosion operation followed by a dilation operation, and the MC operation is a dilation followed by an erosion operation. The dilation operation calculates the maximum pixel value. On the contrary, the erosion calculates its minimum value in a specific SE. Let I be a bit of the greyscale input image and the structure element be SE, the outputs of the MO and MC Oo and OC are:

$$O_o = I \otimes SE \oplus SE \tag{1}$$

$$O_C = I \oplus SE \otimes SE \tag{2}$$

Where $\oplus$ denotes a dilation operation and $\otimes$ denotes an erosion operation. All the structure elements are shaped as rectangles. To efficiently exploit parallelism calculations, we decompose this rectangular shaped structure element into small rectangles in the following hardware implementation phase. As shown in the right hand of the Figure.3, the operation principle of the dilation filter is described as follows. We first pass the current input pixel into the internal "Stage 0" and "Line Buffer 0" simultaneously. Then it is passed to the next stage in the following clock cycles until it reaches "Stage 29". And the maximum pixel value of the current 30 pixels of all the 30 stages is calculated. And the values of the pixels from two consecutive lines of the grey scale image (i.e. 640 pixels per line) are stored into the two line buffers in order to calculate the maximum value from three consecutive pixels from the same column. The structure of the erode filter, as shown in Figure.3, is similar to the dilate filter. The main difference is that the minimum value of the pixels is calculated instead of the maximum one. The erode filter and the dilation filter is executed in Parallel.

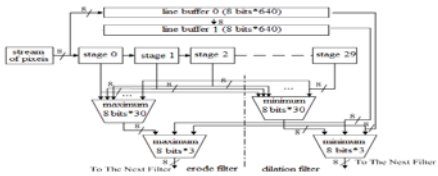


FIGURE III. DIAGRAM OF PIPELINE ERODE AND DILATION FILTER.

The last unit is the CCA module. We select the CCA unit and mark the candidate plate region from the entire binary image. The pixels of the input pixel stream are divided into several blobs by the CCA module, as stated in Figure. 4. The pixels connectivity based grouping is performed as follows. The binary stream is first scanned from left to right starting from the top line. For instance, a comparison between the current pixel "P1" and Fig 4, its upper pixel "P1A" and left pixel "P1L", which have already been grouped, is performed. All pixels with value 0 will be assigned to one group with an index 0. If the value of "P1" is 1 and the indexes of its neighbors are the same and not 0 then "P1" will be assigned the same index as its neighbors. If the indexes of the two neighbors are different and not 0, then the indexes of this pixel and its upper neighbor "P1A" will be the same as its left neighbor (i.e. "P1L"). If the indexes of the two neighbors are different and one of them is 0, then the index of this pixel will be the non-zero index of its neighbor. If the pixel value is 1 and the indexes of its neighbors are both 0, the index of a new group will be assigned to this pixel. Finally, the coordinates of each rectangular shaped group are recorded for candidates selection.

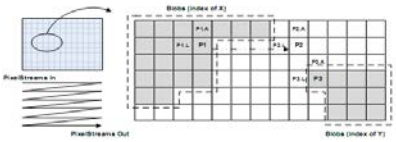


FIGURE IV. The block diagram of CCA.

Once the whole image is scanned, a selection of is performed, which is mainly based on the geometrical relationship. First, considering geometrical factors of the candidate region, that is, the length, width, area and ratio of LP, we define Condition1and Condition2. Condition1 is stricter than Condition 2. Condition1 is adopted to select perfectly suited candidates from good condition images, that is, daytime and clear images. However, Condition 2 can be used for selecting candidates from bad quality images (e.g. night time, blur and complex background images).The maximum and minimum coordinates that pass one of the conditions are returned.

The experiment results show that 96% of the test images met Condition1 and 75% of the remaining images met Condition 2, which means this extra condition can further increase the detection rate by around 3% with no significant increase in the execution time. The validation of the candidates is part of the character segmentation stage and will be addressed in our next work phase.

IV. FPGA IMPLEMENTATION AND RESULTS

The proposed algorithm illustrated in Figure.1 has been successfully implemented and verified using the Mentor Graphics RC240 FPGA development board equipped with a 4M Gates Xilinx Virtex-4 LX40 [13]. The blocks from Figure.1 which have not been discussed in details in this paper (i.e. the RGB to greyscale converter filter, pixels subtraction filter and image binarising filter) have been implemented using Pixel Streams IP cores. The external memory data width is 32 bits, which means every pixel value (24 bits) can be saved on a single memory location. Each RGB pixel is combined with its corresponding position coordinate and synchronization information, which will be sent to the filter blocks in Figure.1 running in parallel, and every clock cycle one data pixel is passed from one block to the next one. A database of 1,000 UK and 1000 Chinese car images with a resolution 640×480 has been used for testing. 1968 LPs out of 2000 (i.e. 98.4%) have been successfully localized.

The original RGB image is first stored in an external memory on the RC240 board. The number of occupied slices available in our system is 18432, among which, our design occupied 521 slices. So the design requires only 2% of the available on-chip resources. This is due to the algorithm needs simple logic, which is easy to be implemented. And the adopted logic here requires no multipliers and no dividers. Moreover, since our design is fully parallelized, the maximum running frequency is 149.120MHz and the total number of clock cycles for one frame is 397740. This means that one LP can be located in 2.7ms. The consuming power of the designed circuits has been analyzed using Xilinx XPower. The obtained total quiescent power and total dynamic power are 386mW, 185mW, respectively. Thus the total FPGA power consumption is 571mW, which is much less than that of a normal PC. Therefore, our design is a low power system.

Moreover, only three Mo and MC operations are involved. This leads to low computational complexity in comparison to other existing methods. Our hardware implementation has been compared with the best two DSP and FPGA implementations in [2] and [12]. The overall performance is displayed in Table 1. Results obtained have shown significant improvements in term of LP localization success rate and consuming time.

TABLE I .COMPARISON WITH EXISTING METHODS.

	[2]	**[12]**	**Our method**
processor	FPGA Virtex II	DSP C6414 and FPGA	FPGA Virtex-4
Clock speed (MHz)	71.062	600	149.120
Image size (pixels)	256×256	352×288	640×480
LP location time (ms)	9.25	141.62	2.7
Success rate of LP location (%)	87	96	98.4

V. CONCLUSION

An efficient LP localization algorithm for FPGA implementation has been proposed. Being on-chip resources of a Virtex-4 FPGA. Adopted parallel computation for FPGA, the system is capable of processing one image in 2.7ms with localization rate of 98.4% at a maximum operation frequency of 149.120MHz. The 2% resources usage of the FPGA for the LP localization implementation leaves 98% of the FPGA area for the remaining parts of an ANPR system, for example, segmentation of LP and character recognition. This enables the entire ANPR system to be implemented on an FPFA that can be placed within an ANPR camera housing to create a stand-alone unit. And this unit will not only drastically improve the energy efficiency, but also remove the installation and cabling costs of bulky PCs situated in expensive, cooled, waterproof roadside cabinets.

ACKNOWLEDGEMENT

This work was supported by National Natural Science Foundation of China No.61102039, Hunan Provincial Natural Science Foundation of China No. 14JJ7029, National hi-tech research and development plan No. 2014AA052600 and the Fundamental Research Funds for the Central Universities.

Yanghong Tan (e-mail: tanyho@126.com) received the M.S. and Ph.D. degrees in electrical engineering from Hunan University in 1997 and 2005, respectively. Currently, he is a Professor with the College of Electrical and Information Engineering, Hunan University, Changsha, China. Her research interests are Power systems and intelligent signal processing.

REFERENCES

[1] F.Lian, Y. Fan, Y. Zhang: Study of tech no logy in electronic toll collection[J] , Compute Engineering and Application, Vol. 43(5), 2007, pp. 204-207.

[2] C. Arth, C. Leistner and H. Bischof: TRIcam: an embedd ed platform for remote traffic surveillance, The IEEE Conference on Comp uter Vision and Pattern Recognition, 2006.

[3] G.A.Smara, F.Khalefah:Localization of License Plate Number Using Dynamic Image Processing Techniques and Genetic Algorithms. IEEE Trans. Evolutionary Computation,2014,Vol.18(2): 244– 257.

[4] X.Zhai, F.Benssali and S. Ramalingam: License Plate Localization based on Morphological Operations. The 11th International Conf. Control, Automation, Robotics and Vision, Dec. 2010:1128- 1132.

[5] Jinn-Li Tan ; Abu-Bakar, S.A.R. ; M.M.Mokji, License plate localization based on edge-geometrical features using morphological approach. 20th IEEE International Conference on Image Processing (ICIP), 2013: 4549 – 4553.

[6] A.H.Ashtari, M.J.Nordin, M. Fathy: An Iranian License Plate Recognition System Based on Color Features. IEEE Transactions on Intelligent Transportation Systems 2014,Vol.15(4) : 1690 – 1705.

[7] F.Kahraman, B.Kurt, and M.G kmen: License plate character segmentation based on the gabor transform and vector quantization, International Symp.Circuits & Systems (ISCAS),2003, pp .381-388.

[8] K. Kim, K. Jung,S. Park, H. Kim: Support vector machines for texture classification , IEEE transactions on pattern analysis and machine intelligence, 2002, Vol. 24, No. 11.

[9] A. Clemens, L. Florian and B. Horst: real-time license plate recognition on an embedded DSP-platform, IEEE Conference on Comp uter Vision and Pattern Recognition, June 2007, p p. 1-8.

[10] I.Trapeznikov, A Priorov, V.Volokhov:Allocation of text characters of automobile license plates on the digital image. Proceedings of 15th Conf. Open Innovations Association FRUCT, 2014 : 144 – 149.

[11] N. Bellas, S. M. Chai, M. Dwyer and D. Linzmeiser: FPGA implementation of a licence plate recognition SoC using automatically generated streaming accelerators, The International Parallel & Distributed Processing Symposium, April, 2006 (ISBN: 1-4244-0054-6).

[12] T. Kanamori, H. Amano, M. Arai, D. Konno, T. Nanba and Y. Ajioka: Implementation and evaluation of a high speed license plate recognition system on an FPGA, The 7th IEEE international Conference on Computer and information Technolog y, 2007, pp. 567-572.

[13] Mentor Graphics, RC240 Datasheet, http://www.mentor.com/, Mentor Graphics Corporation, 2010.