

Compact Pipeline Hardware Architecture for Pattern Matching on Real-Time Traffic Signs Detection

Anh-Tuan Hoang ¹⁾, Mutsumi Omori ²⁾, Masaharu Yamamoto ²⁾, Tetsushi Koide ¹⁾²⁾

Hiroshima University

1) Research Institute for Nanodevice and Bio Systems (RNBS)

2) Graduate School of Advanced Sciences of Matter, Hiroshima University

1-4-2, Kagamiyama, Higashi-Hiroshima, 739-8527, Japan

e-mail : {anhtuan, yamamoto-masaharu, koide}@hiroshima-u.ac.jp

Abstract - This paper describes a novel compact hardware oriented algorithm and its conceptual implementation for real-time traffic signs detection system. The speed limit sign area on a grayscale video frame is detected based on a novel, simple and compact rectangle pattern matching and circle detection modules. The speed limit recognition system is divided into two-pipeline stages. The frame is scanned with multi-scan windows in parallel for each position and each scan windows is also processed in pipeline to increase throughput. It achieves 100% in detection rate.

grain pipeline in order to increase the general throughput with small penalty in hardware size. The traffic sign candidates' position and black-and-white data are then processed for number recognition.

The related works in traffic signs recognition is shown in section 2. The speed limit recognition system architecture and related algorithms for traffic sign candidates detection are shown in section 3. Section 4 and 5 show the conceptual implementation of the introduced architecture and its evaluations.

I. Introduction

The traffic sign recognition would be very important in the future vehicle active safety system. The most important information is provided in the drivers' visual field by the road signs, which are designed to assist the drivers in terms of destination navigation and safe. The most important of a car assistant system is to improve the drivers' safety and comfort. Detecting the traffic signs can be used in warning the drivers about current traffic situation, dangerous crossing, and children path. Although the navigation system is available, it cannot apply to the new roads or the place that the navigation signal cannot reach or in electronic speed limit signs, where the sign changes depend on the traffic condition. An assistant system with speed limitation recognition ability can inform the drivers about the change in speed limit as well as notify them if they drive at over speed. Hence, the drivers' cognitive tasks can be reduced and safe driving is supported. However, meeting real time performance for such a system is still a big challenge research, especially in compact hardware size.

In this study, we aim to solve this challenge and perform real-time speed limit signs recognition on a low resources embedded platform. The targeted platform is the Xilinx Zynq 7020, which has 85K logic cells, 53.2K LUTs, 106.4K registers and 506KB BRAM. In order to get the goal, the fine and coarse grain pipeline together with parallel processing architecture is used. The recognition processing of each frame is coarsely divided into two pipeline stages. One is used for traffic signs and size detection and the other is used for number recognition. The traffic signs and size detection scan each frame by various scan windows sizes in parallel. Each scan windows in one position is finely processed in fine

II. Related Works

In general, most traffic signs systems is structured into pre-processing the obtained frames with various segmentation techniques for signs detection before recognizing and classifying them.

A. Color Based Segmentation

A general feature of the traffic signs is the color. The color is pre-decided to ensure the focus of the drivers. Hence, the color feature can be used as a feature for image segmentation. J. Torresen [1] detects a red circle of the signs by utilizing a Red White Black filter before applying detection algorithm. J. Miura [3] detects sign candidate regions by focusing on the white circular region with some thresholds. F. Zaklouta [7] also uses the color information with enhancement for detection. This detection method required colored camera and more computation resources such as memory for colored image storing and detection.

B. Shape Based Segmentation

Another method for traffic sign candidates detection is based on the shape features of the signs. This approach can be seen in [6], in which the feature that a rectangular structure yields gradients with high magnitudes at its borders is used in the detection. Another research that applies this method is [4], in which the edge detection is used for rectangle detection and Hough-transform is used for circle detection. This method is robust to change in illumination.

However, the available researches required complex computation such as Hough transforms or SIFT feature computation for detection. Calculating the transformation and extracting matching peaks on large image is computationally complex for real-time processing systems.

C. Template Based Matching

Template matching in [1] uses the pre-prepared template for area comparison with various sizes. The approach is simply takes the specific color information of an area and compares with a prepared template for matching. Since the size of the signs is varied from 32x32 to 78x78, a huge hardware resources and computation time are required for the comparison.

III. Compact Hardware Oriented Speed Limit Recognition System with Circle Detection

In order to overcome the weakness in hardware size and computation time of the related work, we would like to introduce a novel algorithm and architecture for traffic sign candidates detection, which are the combination of a simple rectangle pattern matching together with a simple circle detection. The algorithm is shape based segmentation; hence input data are 8-bit grayscale frames. Due to the simplicity of the algorithm, it supports for real-time system.

A. System Overview

Fig.1 shows the speed limit recognition system over views. The system includes two filters: noise reduction filter and sign enhancement filter. A simple noise reduction filter is applied to the input frame for image correction and noise reduction. A sign enhancement filter, which includes sign enhancement and image binalization, is then applied to the output of the noise reduction filter, changing the 8-bit value

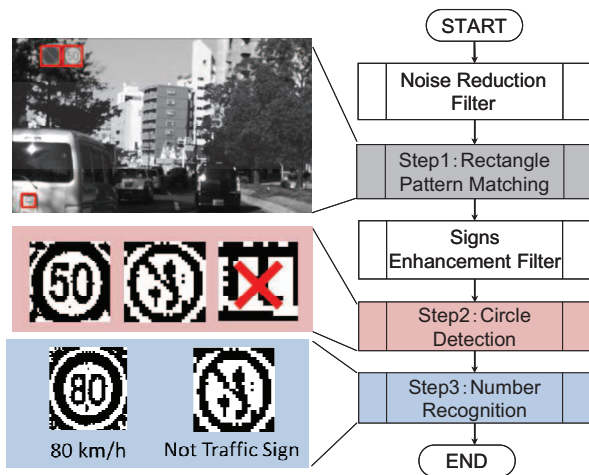


Fig. 1. Speed limit recognition system.

grayscale pixels to 1-bit value black-and-white pixel for number recognition. The sign enhancement process helps increase the features of the numbers and reduces the amount of data being processed. Rectangle Pattern Matching step roughly detects the traffic sign candidates based on the difference in the brightness between the circle line and the neighbors. The circle detection reduces the number of traffic sign candidates detected by rectangle pattern matching, decides if the detected regions of interest (ROIs) are really a circle mark or not. Finally, the number recognition module analysis features of the ROIs and compares with the features of the numbers for number recognition.

Fig.2 shows examples of ROIs detected in the real life in highways and local roads by the proposed system. A scene is considered as all frames that the same sign appear in the observable field of the camera until it disappears from the observable field. In the same scene, when the camera is far from the sign, size of the sign is as small as 16x16 pixels. This size increases to 39x39 pixels when the camera gets closer to the sign. Our proposed traffic signs detection algorithm is designed to recognize the size in a scan window (SW) in a range of 15x15 pixels to 50x50 pixels. In the real implementation, the unnecessary SW size is removed to reduce hardware size.

B. Noise Reduction Filter and Sign Enhancement Filter

The noise reduction filter is used to reduce noise of the input image.

Sign enhancement filter in our implementation is a special filter for sign enhancement and image binalization purposes. It is used to strengthen the features of the image and also reduces the amount of processing data at next step. Coefficients of the noise reduction filter and sign enhancement filter are specially selected for hardware



Fig. 2. Examples of ROIs detected in the real life in daily condition at different distances.

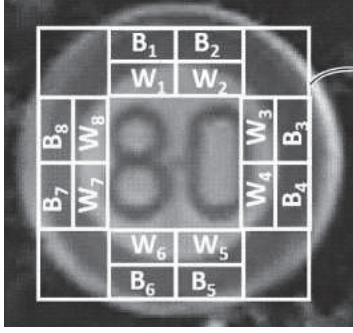


Fig. 3. Rectangle pattern matching for traffic sign detection.

oriented implementation purpose.

C. Rectangle Pattern Matching

Fig.3 shows how the rectangle pattern matching can be used for circle detection in the traffic signs recognition. The scan window corresponds with the traffic sign is processed in 16 areas, from B1 to B8 (says for Black) and W1 to W8 (says for White). The differences in brightness of corresponding Black and White areas (B1 and W2, B2 and W2, etc.) are computed. If they get over a threshold, the pattern is considered as a rectangle. At different view points, the brightness of the black and white areas is different. In our implementation, the threshold is considered as a variable, which relies on the different in brightness between W1 and B1.

D. Circle Detection with Improved Border Patterns

The border templates of a circle are used to decide if the detected rectangle is a circle or not as shown in Fig.4. In our implementation, we improve the original matrix of 2x2 border templates to matrixes of 3x3 for inner-circle and outer-circle lines detection as shown in Fig.5. This improvement increase the accuracy of traffic speed signs detection but also provides some penalty on hardware size for the detector.

IV. Compact Conceptual Pipeline Architecture



Fig. 4. Voting for circle using border templates.

Inner-circle line detection

3x3 template for inner-circle detection						
Direction	↓	↑	←	→	↘	↙

a. Improved border template for inner-circle detection.

Outer-circle line detection

3x3 template for outer-circle detection						
Direction	↓	↑	←	→	↘	↙

b. Improved border template for outer-circle detection.

Fig. 5. Improved border template for circle detection.

A. System Pipeline Architecture

Fig.6 shows the pipeline architecture of the speed limit recognition system. The system is able to scan for the traffic signs up to 50x50 pixels in size. The input image is 8-bit grayscale 640x390=249,600 pixels.

It contains two main modules of Rectangle Pattern Matching (RPM) and the Number Recognition (NR). Other modules are Noise Reduction Filter, Sign Enhancement Filter, and Circle Recognition as shown in section III. Supported for the Noise Reduction Filter and the RPM are a number of 8-bit FIFOs.

The Sign Enhancement Filter is independent with the RPM processing, and so could be processed in parallel with the RPM. Different from the software implementation, in which

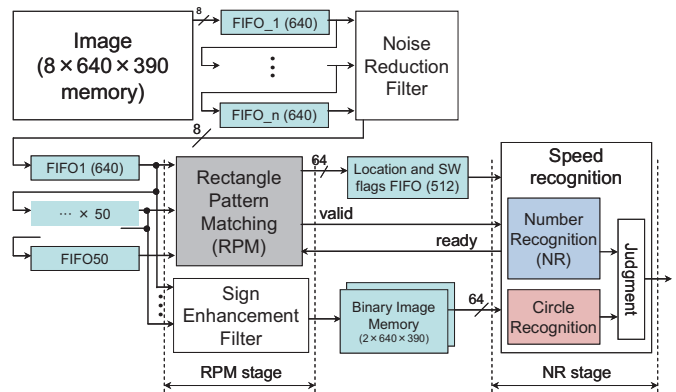


Fig. 6. Speed limit recognition system with scan windows sized up to 50x50 pixels.

the Circle Recognition is used to reduce the number of sign candidates, in the hardware implementation, the circle recognition result is used to strengthen the judgment of the speed limit recognition. Hence, the Circle Recognition and Number Recognition can be executed in parallel before the final judgment. The Noise Reduction Filter, the RPM, and the Sign Enhancement Filter work with 8-bit grayscale data while the NR and the Circle Recognition work with 1-bit black-and-white data. These 8-bit processing modules and 1 bit processing modules are connected with the other through two memories. The first one, the Location and Scan Windows flags FIFO, is used to store the position of the sign candidates in a frame and the detected scan window sizes at that position. The second one, a general memory called Binary image memory with the size of 249,600 bits, is used to store the black-and-white bit value of each frame. Two independent memories and a memory swapping mechanism are necessary. It allows the 8-bit and 1-bit processing parts access the binary image memories for read and write in parallel.

Fig. 7 shows the two pipeline stages, named RPM and NR, of the speed signs recognition system. The Noise Reduction Filter, the Rectangle Pattern Matching, and the Sign Enhancement Filter occur at the RPM stage. The RPM and Sign Enhancement Filter works in parallel using data generated from the noise reduction filter. The scan windows (SW) in RPM and Sign Enhancement Filter are pipeline processed with one input pixel at each clock. Hence, about $390 \times 640 = 249,600$ clocks are necessary for the first stage. During the processing time of the first frame, the detection result is written into the result FIFO and the binarization image result is written into the Binary Image Memory 1 for the next stage. At the next stage, the Circle Detection and the Number Recognition modules read data from the FIFO and the Binary Image Memory 1 for processing before handling result to the Judgment module. At the same time, the data of the second frame is processed in the RPM stage. The result is written into the Binary Image Memory 2. Then, the NR stage of the second frame occurs with the previous written data inside the Binary Image Memory 2. The same process occurs with other frames, and so the system processes all frames in pipeline.

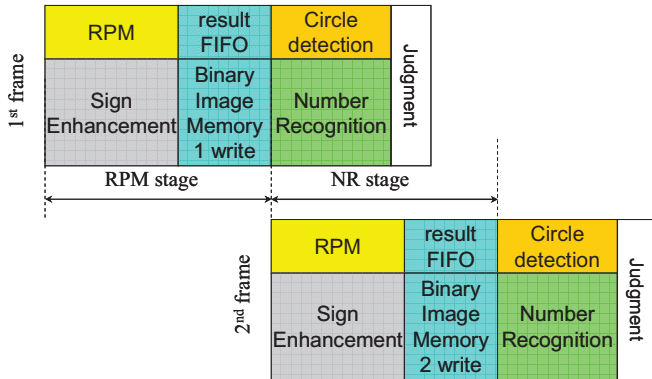


Fig. 7. Pipeline stages of speed recognition system.

B. Rectangle Pattern Matching Logic Design

There are two reusable computation are applied to the rectangle pattern matching to reduce the hardware size. The first one is globally applied to the scan windows inside a frame as shown in Fig.8. During the processing of scan window 1 (SW1), the brightness of area W3, W4, B3, and B4 are computed. During the SW_n, these areas became B7, B8, W7 and W8, respectively. Hence, the brightness computation results for those areas in SW1 can be stored for reuse in SW_n. The second re-usable computation is locally applied to the brightness computation of two continuous areas as shown in Fig.9. The overlapped area between S1 and S2 is reused without computation. The computing area S2 is generated by the overlapped area plus the new input area (S_{add}). The overlapped area is the computed area S1 subtracts for the subtraction area (S_{sub}). The computation now became computing for the addition area (S_{add}) and storing the result for latter use. At the same time, the newly computed S_{add} area is added with S1 before subtracting the previous stored S_{sub} area for S2 computation.

Fig.10 shows the logic for B1 area computation using the re-usable method introduced in Fig.9 The input data corresponds with the S_{add} area. The brightness of this area are

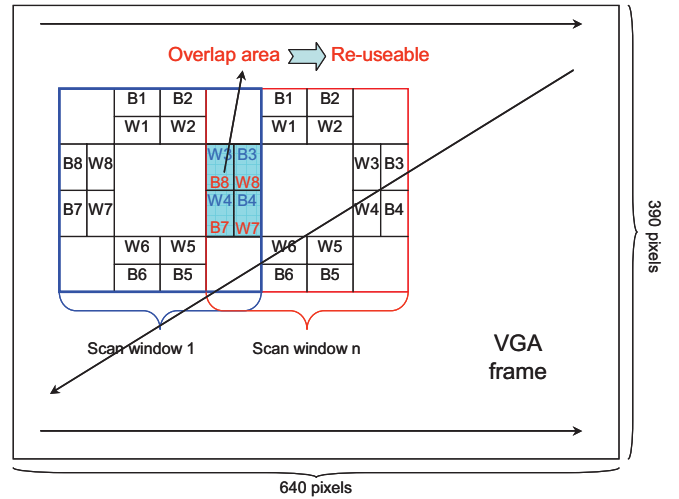


Fig. 8. The overlap and re-usable computation result in different scan windows in a frame.

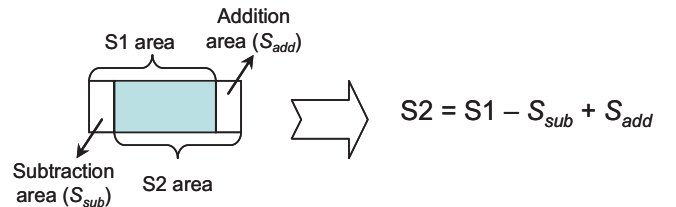


Fig. 9. The overlap and re-usable result in two continuous areas.

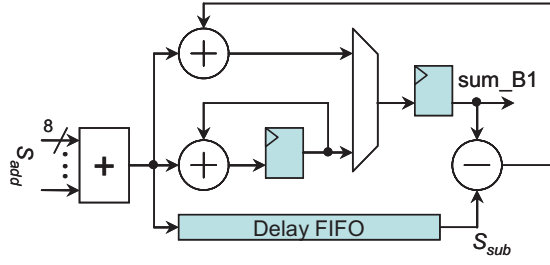


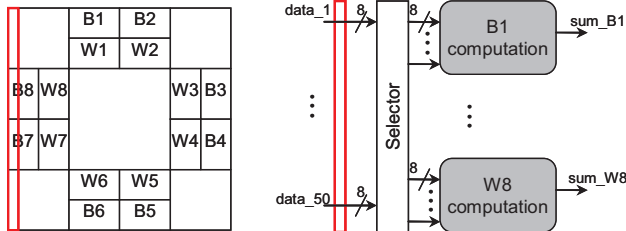
Fig. 10. Computation of an area using previous result.

added together and stored to the delay FIFO for latter use as S_{sub} area. The previous computed area B1 is stored in the output register. The addition and subtraction of the S_{add} and S_{sub} areas from B1 generates the next B1 value and stored to the output register. One more register is used for the initial computation.

The general overview of the input data in a virtual scan window is shown in Fig.11.a and how those data are read in to processing logic is shown in Fig.11.b. At each clock, data of 50 pixels in a column are read and directed to the B1 ~ W8 computation modules. Design for the brightness computation for each area of B1 ~ W8 is shown in Fig.10. The computation for other smaller scan windows size (smaller than 50x50) at the same position can be done in parallel with the same 50 input pixels in a column.

C. Circle Detection Logic Design

The circle recognition logic processes with each traffic sign candidate to check if it has the circle form or not. Hence, input data to the module is the SW candidate at a specific position. Depend on the location of the pixel inside the input SW, the direction of the templates in Fig.5 are identified. Hence, we simply make comparison between the detected direction of the input pixel and the expected direction at that location and count the number of matches. If the number of matches gets over a threshold, the input SW is considered as a circle.



a. 50 input data in a visual 50x50 scan window. b. 50 input data in physical logic for rectangle detection.

Fig. 11. Data in a 50x50 scan window processing for rectangle detection.

Fig.12 shows the mechanism and design of the circle recognition module. The 3x3 array is used to detect the direction of the input pixel using the border templates introduced in section III.D. The direction is then compared with the expected direction for that pixel. Number of matches is computed by the adders and stored into the register. The final number of matches is compared with a previously decided threshold. If the number of matched is bigger than the threshold, the input SW is considered as a circle.

D. Number Recognition Algorithm

In our research, the number recognition algorithm is a specific designed algorithm. A number is recognized through the features of histogram and continuity of black and white pixels. It is the subject of our other paper [9].

V. Discussion on the Architecture Implementation

In terms of detection accuracy, the simulation has been done for ROIs detection in 54 real life scenes in normal day light on highways (31 scenes) and local roads (23 scenes). However, the electronic speed limit signs are not included and are subject of the next step. One scene is considered as all frames that a traffic sign is first appeared in the observation field of the camera until it disappears. Those scenes are taken under various light conditions such as normal light, backlit, and on tunnel. A sign in a scene is considered to be detected if it is identified in any frame of that scene. In all the normal scenes, our proposed algorithm successfully detects the ROIs for the traffic signs. There are some difficult situations that need to improve. Fig. 13 shows a difficult situation occurs under the backlit condition, in which the algorithm correctly recognizes the previous frame but fails to recognize the next one. Another unsuccessful situation occurs in Fig. 14 due to the bad quality of the sign.

In terms of hardware size for implementation, this architecture processes SW in pipeline. Data of a whole SW does not required at the same time, and so, reduces a huge

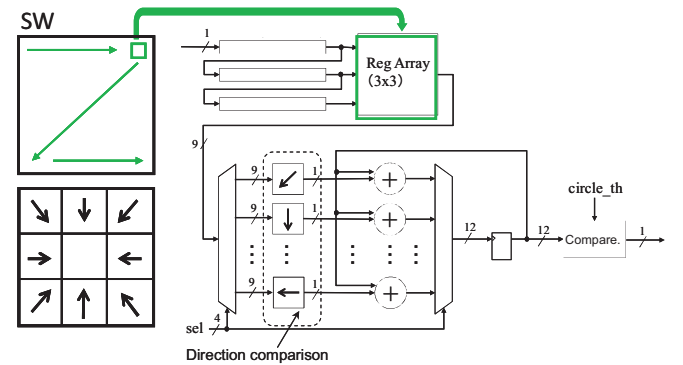


Fig. 12. Mechanism and design of the circle detection module.



a. Detectable frame b. Undetectable frame

Fig. 13. Two continuous frames in the same scene under backlit condition.



Fig. 14. Undetectable due to the quality of the sign.

number of registers. In more detail, a straight forward implementation of a SW with 50×50 pixels in size requires $50 \times 50 \times 8 = 20,000$ registers. In our proposed pipeline method, the number of registers reduces to $50 \times 8 = 400$ registers only. In terms of SW size, it is not necessary to scan all 30 SW sizes in the size range of 15×15 pixels to 50×50 pixels. Instead, the simulation results show that two SWs among them are enough for traffic speed signs detection and recognition. The significant decrease in the number of registers and number of SWs allows us to implement those multiple SW sizes in parallel to increase the throughput while maintaining the compact size in terms of hardware size. The computational re-use mechanism introduced in section IV also helps us significantly reduce the computation logics and shorten the critical path. In estimation, the system with two SWs processing in parallel occupies 8,168 registers and 26 BRAMs, and so be able to be implemented on Xilinx Zynq 7020 system, which has 106,400 registers and 140 BRAMs. The critical path in the RPM is very short with 8-bit 9-input adder, a multiplexer and a comparator (for the largest 50×50 SW size). One SW can be completed in one clock; Short critical path with small hardware size allows the algorithm to be implemented on Zynq for real time system.

VI. Conclusion

This paper introduces our novel algorithm and implementation for traffic board sign candidate detection. The combination of coarse and fine grain pipeline architectures with parallel processing allows our implementation meets the demand of real time system. The

two computation re-use mechanisms in scanning process significantly reduce the hardware size in our implementation to 1/50 compared with the straight forward scanning method and allow compact hardware size implementation.

In the future, the algorithm will be implemented and verified on FPGA. In addition, the algorithm will be tested with other environments such as LED signs and signs under backlit condition.

Acknowledgements

Part of this work was supported by Grant-in-Aid for Scientific Research(C) JSPS KAKENHI Grant Number 23560400.

References

- [1] J. Torresen, et al., "Efficient Recognition of Speed Limit Signs," *Proceeding of the 7th International IEEE Conference on Intelligent Transportation Systems*, pp. 652-656, 2004.
- [2] P. M. Ozelik, et al., "A Template-Based Approach for Real-Time Speed-Limit-Sign Recognition on an Embedded System Using GPU Computing", *Proceedings of the 32nd DAGM conference on Pattern recognition*, pp. 162-171, 2010.
- [3] J. Miura, et al., "An Active Vision System for Real-Time Traffic Sign Recognition", *Proceeding of International IEEE Conference on Intelligent Transportation Systems*, pp. 52-57, 2000.
- [4] F. Moutarde, et al., "Robust on-vehicle real-time visual detection of American and European speed limit signs, with a modular Traffic Signs Recognition system", *Proceeding of International IEEE Intelligent Vehicles Symposium*, pp. 1122-1126, 2007.
- [5] Peemen, et al., "Speed sign detection and recognition by convolutional neural networks", In *International automotive congress*, 2011.
- [6] C. G. Keller, et al., "Real-time Recognition of U.S. Speed Signs", *Proceeding of International IEEE Intelligent Vehicles Symposium*, pp. 518-523, 2008.
- [7] F. Zaklouta and B. Stanculescu, "Segmentation Masks for Real-time Traffic Sign Recognition using Weighted HOG-based Trees", *Proceeding of the 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 1954-1959, 2011.
- [8] G. K. Siogkas and E. S. Dermatas, "Detection, Tracking and Classification of Road Signs in Adverse Conditions", *Proceeding of International IEEE on Electrotechnical Conference*, pp. 537-540, 2006.
- [9] M. Yamamoto, et al., "Speed Traffic-Sign Recognition Algorithm for Real-Time Driving Assistant System", *Proceeding of the 18th Workshop on Synthesis And System Integration of Mixed Information Technologies (SASIMI 2013)*.