

RESEARCH

Open Access

Design and implementation of a real time and train less eye state recognition system

Mohammad Dehnavi* and Mohammad Eshghi

Abstract

Eye state recognition is one of the main stages of many image processing systems such as driver drowsiness detection system and closed-eye photo correction. Driver drowsiness is one of the main causes in the road accidents around the world. In these circumstances, a fast and accurate driver drowsiness detection system can prevent these accidents. In this article, we proposed a fast algorithm for determining the state of an eye, based on the difference between iris/pupil color and white area of the eye. In the proposed method, vertical projection is used to determine the eye state. This method is suitable for hardware implementation to be used in a fast and online drowsiness detection system. The proposed method, along with other needed preprocessing stages, is implemented on Field Programmable Gate Array chips. The results show that the proposed low-complex algorithm has sufficient speed and accuracy, to be used in real-world conditions.

Keywords: eye state, pupil, iris, drowsiness, vertical projection, FPGA

Introduction

All over the world and every day, driver's fatigue and drowsiness have caused many car accidents. In fact, drowsiness is the case of about 20% of all car accidents in the world [1,2]. As a result, an electronic device to control the driver's awareness is needed. This device should monitor and detect the driver's drowsiness online and activate an alarm system immediately.

In recent years, many researches on these systems have been done and their results are reported [3-12]. One of these methods is to monitor the movement of the vehicle to detect drowsiness of the driver [3]. This method depends very much to the type of vehicle and the condition of road. Another method is to process the electrocardiogram (ECG) signals of driver [4]. In this system, some ECG probes are needed to be connected to the driver, which are disturbing the driver. There are other methods based on processing of the image of driver's face and eye. Some of methods in this category are to process the image of driver and to monitor his/her eye blinking [5-11]. In these systems, the face process, eye region detection process, and eye state recognition process are performed.

In order to determine state of an eye, authors of [5] propose a method based on combination of projection and the geometry feature of iris and pupil. Authors of [6,7] use the fact that the iris and pupil are darker than skin and white part of the eye. Authors of [11] proposed an algorithm based on the cascade AdaBoost classifier. In [12], a gray level image of an eye is converted to a binary image, using a predetermined threshold. Then, based on the number of black and white pixels of this binary image, state of the eye is determined.

The algorithm presented in [8] used the Hough Transform to detect the iris and to determine openness of the eye. Authors of [13] used three steps to recognize the eyes' state. In the first step, the circular Hough transform is used to detect the circle of an iris in the image of an open eye. If this circle is not found then in the second step, the direction of the image of upper eyelid is obtained to determine whether it is below of the line between two corners of an eye, to detect a closed eye. If a closed or open eye is not determined in the first two steps, then in the third step, the standard deviation of distance between upper and lower eyelids is obtained and is compared to a threshold to determine the eye state.

Some researches are based on the projection of the image, to determine the state of an eye. In [9], the

* Correspondence: mo.dehnavi@mail.sbu.ac.ir
ECE Department, Shahid Beheshti University, Tehran, Iran

vertical projection of the image of both eyes is used. In [10], horizontal projection image of an eye is used to determine the interval between eyebrows and eyelids and to recognize the state of an eye. In [14], the horizontal projection of the image of a face is calculated to determine state of an eye.

Some works also are based on “Supported Vector Machine” (SVM) classifier. In [15], the SVM classifier is used to detect state of the eye. Authors of [16] used SVM classifier and Gabor filter to extract eye characteristic.

In the above methods, the authors used some conditions which make some difficulties in the eye state recognition. For example, the algorithm presented in [5] has many stages which make it slow. As a result, this method cannot be used in a real-time system. Conditions such as light from different angles, dark eyelashes, eyebrows image located in eye block, and glasses decrease the accuracy of algorithms presented in [6,7].

Since Hough transform has a massive calculation, the algorithm proposed in [8] is also slow. Algorithm in [9], in addition to higher computation, has a high sensitivity to light radiation. Some factors such as difference interval between eyebrow and eyelid, changes of environment light, and the color of eyebrow and eyelash highly affect the accuracy of the proposed algorithm in [10]. Algorithms in [11,15,16] have a training phase and also their hardware implementation is complicated. Determining the threshold of algorithm presented in [12] is difficult and this algorithm is very sensitive to the light condition. Algorithm in [13] has also many stages and hardly can be implemented on a hardware platform. The accuracy of the algorithm proposed in [14] is not enough to be used in driver’s situation.

In this article, a new algorithm to recognize the state of an eye, without constraints of the previous methods, is proposed. This algorithm has less sensitivity to the light conditions than other algorithms, with no need to a training phase. In order to verify the correctness of the proposed algorithm, a computer simulation is developed. In order to check and compare the speed of the proposed algorithm, we implemented it on a Field Programmable Gate Arrays (FPGA) hardware platform. The results show a fast performance and acceptable accuracy for the proposed train less eye state recognition algorithm.

The rest of this article is organized as follows. In Section 2, our real-time algorithm to determine an open or closed eye is described. The computer simulation results of the proposed algorithm are provided in Section 3. In Section 4, the design of hardware implementation of the proposed algorithm is presented. The result of this hardware implementation is provided in Section 5. Comparisons between our algorithm and others are presented in

Section 6. In Section 7, the conclusion of article is presented.

Real-time eye state recognition

In a driver drowsiness detection system based on image processing, first the location of the face in the image is determined. Then, place of eyes are determined and finally the image of an eye is processed to recognize the state of the eye. The overall driver drowsiness detection system is shown in Figure 1. In this article, it is assumed that the face detection and localization of eyes are performed with one of the methods presented in [17-22]. Our proposed algorithm recognizes the state of eyes to determine the driver drowsiness.

The proposed algorithm is as follows: First the gray level image of an eye is captured. Then, the vertical projection of this image is obtained, by adding the gray level of pixels in each column. For an $m \times n$ image the vertical projection vector, PV, is calculated using below equation.

$$PV(j) = \sum_{i=1}^m IM(i, j), \quad \text{for } j = 1 \text{ to } PVlen \quad (1)$$

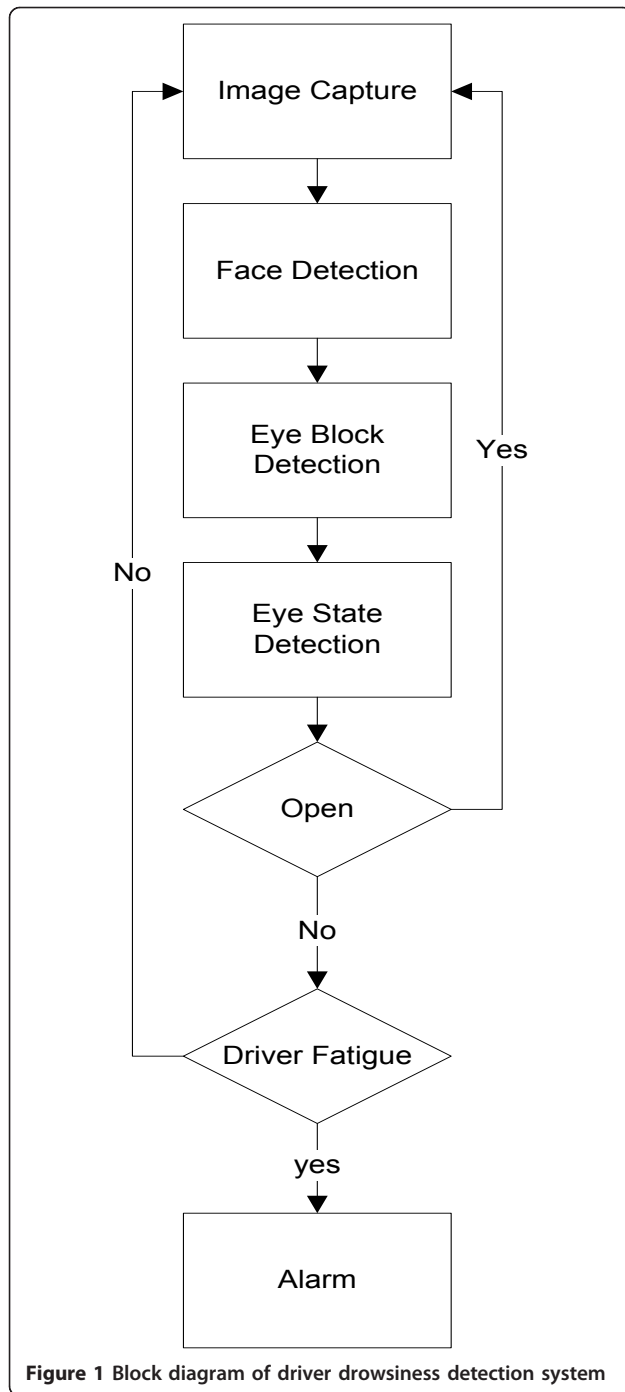
where i is the row number, j is the column number, and $PVlen = n$ is the size of this projection vector. For example, the original vertical projection of an image of an eye shown in Figure 2a is depicted in Figure 2b.

The vertical projection vector needs to be smoothen. To obtain a smooth vector, we use an averaging filter. The size of this averaging filter, AFlen, is considered to be the floor of $PVlen/7$ (Equation 2).

$$AFlen = \text{floor}(PVlen/7) \quad (2)$$

As shown in Figure 2a, the image of an open human eye has three different areas, pupil/iris in the middle and two white parts in the left and right sides. However, in the image of a closed eye, these areas are not discriminated. The effect of this observation in the projection vector is the basis of our proposed algorithm to determine the state of the eye. As shown in Figure 2b, the projection vector of an eye in the open state has three areas. The projection vector of the pupil and iris area has less gray values than two other white areas. As a result, the projection vector has a local minimum in its middle, belongs to pupil area, and two local maximums, belong to two white parts of the eye.

The method which searches for these local minimum and maximums in the projection vector is as follows. First, we add $AFlen/2$ zeros to left and right sides of projection vector, to generate the zero padded projection vector (ZPPV), with a length of $ZPlen = PVlen + AFlen$. Then, the local maximums and



minimums of this vector are obtained. The local minimums that are located between the two maximums are classified in different groups. Each minimum and two adjacent maximums form a group. In each group, the minimum is occurred at X_{min} of zero padded projection vector with a value of Y_{min} . In each group, also the smallest maximum is at X_{smax} of zero padded projection vector with a value of Y_{smax} . If at

least one group in the ZPPV of the image satisfies the both following conditions then the eye is open, otherwise it is closed.

Condition-1: The ratio of difference between Y_{smax} and Y_{min} to Y_{smax} is greater than a threshold value, θ ,

$$\frac{Y_{smax} - Y_{min}}{Y_{smax}} > \theta \quad (3)$$

Condition-2: The minimum that satisfies *condition-1* is located almost in the middle of ZPPV. That is, location of this minimum, X_{min} is between $0.4ZPlen$ to $0.6ZPlen$.

$$0.4ZPlen < X_{min} < 0.6ZPlen \quad (4)$$

The ratio stated in *condition-1* is based on the difference between the color of the pupil (black) and the white area of an eye. This difference varies when the state of an eye is changing from an open state to a closed state. That is, when an open eye is going to be closed it passes different steps. *Condition-1* verifies that an eye is open when this ratio is greater than a threshold value. In the other hand, in the relaxed and open eye, such as driving situation, the pupil almost in the center of eye, *Condition-2* checked this condition to validate the openness of the eye.

As an example, consider image of an open eye, as shown in Figure 2a. Figure 2b is its projection vector and Figure 2c shows the smoothed and ZPPV of this image. Based on our experiments the threshold of *Condition-1* is considered to be $\theta = 0.05$. Figure 2c is satisfied both conditions of an open eye, therefore it belongs to an open eye.

As another example, consider Figure 3a, Figure 3b is its projection vector, and Figure 3c is the ZPPV of this image. This ZPPV is not satisfied *Condition-2* of proposed algorithm; therefore this image belongs to a closed eye.

In the RGB color image of an eye, the red color of the iris of dark and bright eyes are almost similar [23]. Therefore, in our proposed method, the RED component of the RGB color image is used. As a result, the effect of eyes color in the image is declined. The proposed algorithm is shown in Figure 4.

Simulation results

To test the proposed method, we used 450 images of eyes from The Caltech Frontal Face Database [24]. This database contains different image dimensions, different light conditions, and different states of the eye for different people. The computer simulation of the proposed algorithm is run on all images of this database. The results of this simulation are shown in Table 1. Our proposed algorithm shows a 89.5% accuracy to

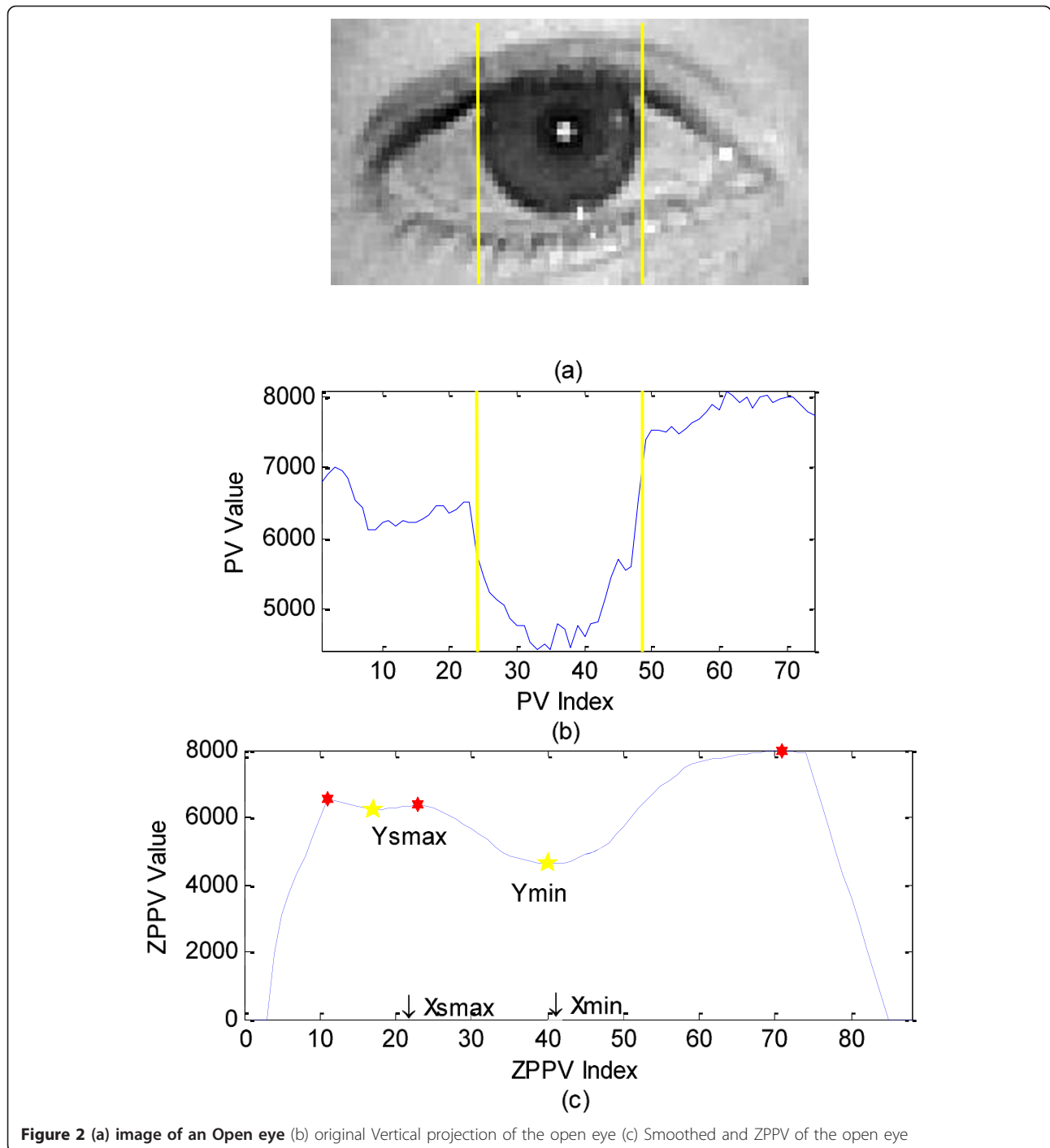


Figure 2 (a) image of an Open eye (b) original Vertical projection of the open eye (c) Smoothed and ZPPV of the open eye

detect open eyes. It also shows 81.8% accuracy in the processing of closed eyes. In total, the accuracy of the proposed system to detect the state of eyes reaches 89.3%.

For further test of the proposed algorithm, we captured 70 images of the eyes in difference conditions in our laboratory. Then, we run test simulation of the

algorithm to the images of Eye SBU database [25]. The result obtained from these images is shown in Table 2. These results show that the proposed system has more than 89% accuracy. These results are almost the same as the results presented in Table 1. It is worth mentioning that this accuracy is obtained through the presented system without any training phase.

Hardware implementation design

In order to implement the proposed algorithm on a hardware platform, we assume that the image of an eye is stored in a Random Access Memory (RAM). In this implementation, we use a RAM with 136×82 bytes, called IMAGE_RAM, to store an image. We also used a True dual port RAM, called PV_RAM, with 136 of 15-bit words, to store the projection vector. Three other major units in the design are data smoothing unit, local max/min search unit, and condition checking unit. These units are controlled through a control circuit. The schematic of this design is shown in Figure 5.

Vertical projection vector is obtained in the first part of this system containing IMAGE_RAM, PV_RAM, and Adder1. All elements of each column of the image, IMAGE_RAM, are added together, by ADDER1, and the result is stored into vertical projection vector, PV_RAM, through port B. the stored Data is read from PV_RAM through port A. This data corresponds to a column of

the eye image, as shown in Figure 5. The PV_RAM and its connection are shown in Figure 6a.

The address of IMAGE_RAM is generated through IMAGE_RAM_ADRESS register/counter, a ring counter which counts from 0 to 11151 (Figure 6b). The address of PV_RAM is also generated by PV_RAM_ADRESS1, a ring counter which counts from 0 to 135.

When the projection vector is completed, PV_C flag is set; and then smoothing unit starts its process, as shown in Figure 6b. For implementation of smoothing unit, we use the following procedure:

Since the length of the projection vector is 136, the length of smoothing filter is equal to 19 (Equation 2). The values of all tap weights of this smoothing filter, averaging filter, are considered to be '1'. In the smoothing process, there are three distinguish phases. In the first phase, smoothing filter has some overlap with data projection vector from the left. In the second phase smoothing filter is completely overlapped with data projection vector. In the third phase, smoothing filter has

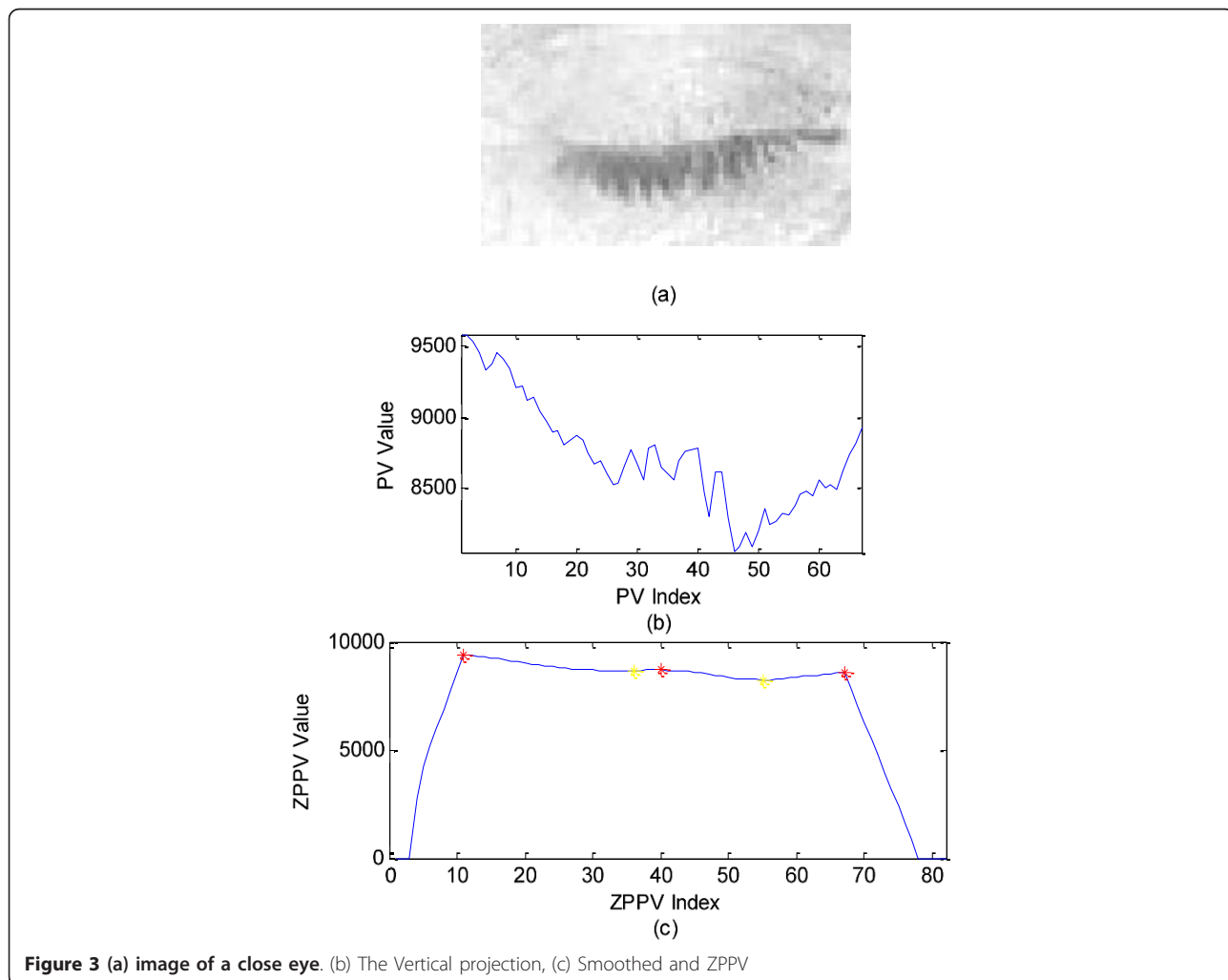


Figure 3 (a) image of a close eye. (b) The Vertical projection, (c) Smoothed and ZPPV

- 1) Capture image of an eye.
 - 2) Calculate the vertical projection vector.
 - 3) Add $AFlen/2$ zeros to the left and right sides of projection vector to generate zero padded projection vector (ZPPV) with length of $ZPlen$.
 - 4) Smooth zero padded projection vector.
 - 5) Obtain maximums and minimums of ZPPV.
 - 6) Place each minimum and two adjacent maximums in a group. In each group, the minimum is occurred at $Xmin$ with a value of $Ymin$. In each group also the smallest maximum is occurred at $Xsmax$ with a value of $Ysmax$.
 - 7) If in one of above groups two following conditions are satisfied then the eye is open, otherwise eye is closed.
- Condition-1: $\frac{Ysmax - Ymin}{Ysmax} > \theta$
- Condition-2: $0.4ZPlen < Xmin < 0.6ZPlen$

Figure 4 The proposed algorithm to detect the state of an eye

some overlap with the projection vector from the right. Figure 7 shows these three phases.

In smoothing procedure, we need to divide the summation of projection vector data which are overlap with smoothing filter to the length of the smoothing filter, $AFlen$. To simplify the hardware implementation of this procedure we approximate the $AFlen$ with the nearest number of 2^k , less than $AFlen$. In the other word, if $2^k < AFlen < 2^{k+1}$ then the denominator consider to be 2^k . Dividing a number by 2^k is a k -bit shift to the right. In our simulation, for example, since $AFlen = 19$, we considered $k = 4$ and instead of division in averaging procedure, we shift the result 4 bits to the right.

Smoothing flag, SC_F , is set during smoothing procedure. Figure 8 shows the architecture of the smoothing unit.

In Max/Min searching unit, the local maximum and minimum of smoothed vector, generated through smoothing unit, are obtained. The input to this unit is the arrays of smoothed vector which is generated at each step. In order to find the local minimum and

maximums of the smoothed projection vector, we used the following method.

In this method, each element of smoothed vector, d_i , is compared to the previous element, d_{i-1} . If $d_i \geq d_{i-1}$ then a '1' is shifted into a 9-bit shift register. This shift register always contains the result of the last nine comparisons. If this shift register has a pattern equal to "000001111" then it indicates a maximum in the smoothed vector. If this shift register has a pattern equal to "111100000" then it indicate a minimum in the smoothed vector. Otherwise, there is neither a minimum nor a maximum in this part of the vector.

When a maximum or minimum is found, its type (max/min), its location (the index of the vector), and its value of the smoothed vector at this minimum or maximum, d_{i-4} , are stored. If a maximum occurred then a '1' is stored in the type register, otherwise, a '0' is stored. The index of the location is stored in the step counter. The value of the smoothed data of this minimum or maximum, d_{i-4} , is obtained from R4 of the register bank. Figure 9 shows the architecture of this process.

Last unit of this system is the condition checking unit. In this unit, conditions of maximums and minimums are checked, concurrently. In each step of type checking, three successive bits of type register is selected. If these three bits have a pattern of "101" then a minimum exist between two maximums. We called this minimum in this group as $Ymin$. The index of this minimum in ZPPV is called $Xmin$. Also in each group one of the maximum is less than the other which we called it $Ysmax$.

If $Xmin$ in Equation (4) is between (54,81), then *Condition-2* is satisfied.

Concurrently, *Condition-1* is checked as follows: first, rewrite Equation (3) as Equation (5).

$$Ysmax - Ymin > \theta \times Ysmax \quad (5)$$

We set $\theta = 0.05$; hence, Equation (5) is rewritten as Equation (6).

$$Ysmax - Ymin > 0.05 \times Ysmax \quad (6)$$

Where in binary mode we have Equation (7)

$$Ysmax - Ymin > (2^{-5} + 2^{-6}) \times Ysmax \quad (7)$$

Table 1 Results of computer simulation in The Caltech Frontal Face Database

	Number of corrects	Number of incorrect	Accuracy (%)
Open eye	393	46	89.5
Close eye	9	2	81.8
Total	402	48	89.3

Table 2 Results of computer simulation in SBU database [25]

	Number of corrects	Number of incorrect	Accuracy (%)
Open eye	32	4	88.8
Close eye	30	4	88.2
Total	62	8	88.7

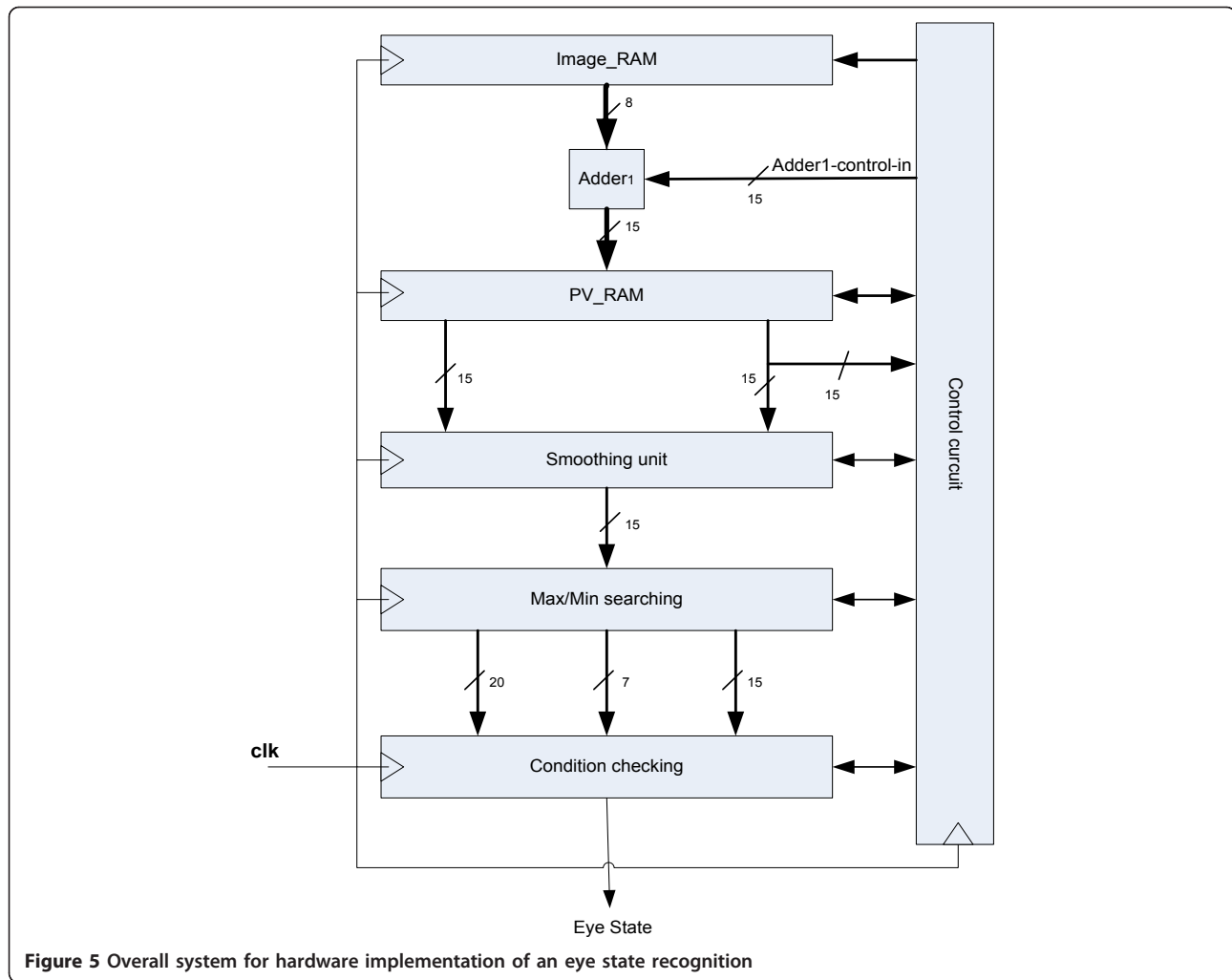


Figure 5 Overall system for hardware implementation of an eye state recognition

Both sides of Equation (7) are multiplied by 2^8 to obtain Equation (8).

$$2^8 \times (Y_s \max - Y_s \min) > (2^3 + 2^2) \times Y_s \max \quad (8)$$

Equation (8) is simpler than Equation (2), and easier for implementation. To implement this equation, at the beginning, two maximums are compared. The minimum of these is $Y_s \max$. To calculate the left side of Equation (8), the Y_{\min} and Y_{\max} are subtracted and the result is shifted to left by 8-bits. Concurrently, $Y_s \max$ is shifted 2-and 3-bits to the left and they are added to obtain right side of Equation (8). The comparator unit compares these two sides to check the threshold condition. Figure 10 shows the architecture of this unit.

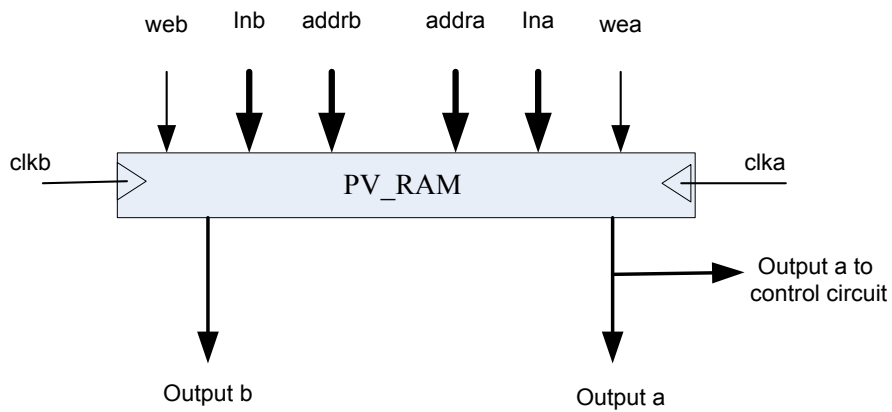
This procedure is repeated for all frames and **PV_RAM** is cleared when the process of one frame is completed. Therefore, data are overwritten on **PV_RAM**, when **IMAGE_RAM_ADDRESS** points to the first column of **IMAGE_RAM**. Control circuit controls

the flow of data. When the first row of **IMAGE_RAM** is read then the inputs of the adder are connected to **IMAGE_RAM** and '0'. When the other rows of **IMAGE_RAM** are read then the inputs of the adder are connected to **IMAGE_RAM** and port B of **PV_RAM**. Figure 6b shows this operation.

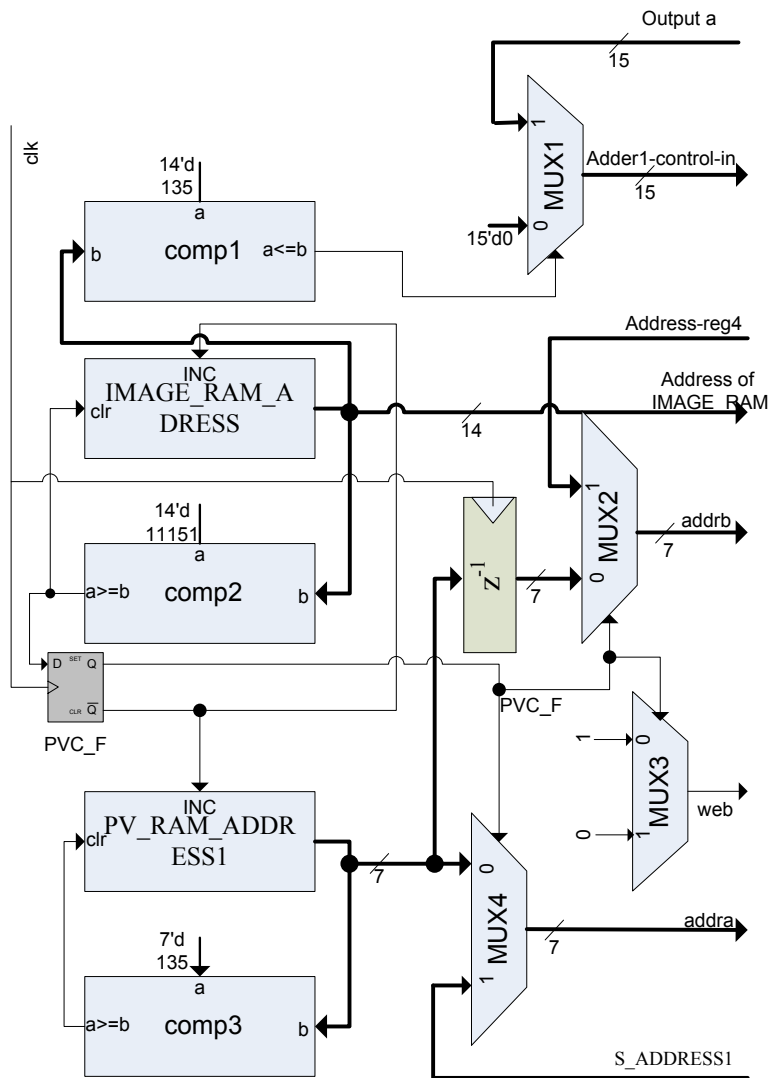
The total needed clock cycles to complete an eye state recognition in an 11152 pixels image is 11310 clocks, 11153 clocks to obtain projection vector, 137 clocks to obtain smoothed data; and 20 clocks to check the conditions.

Hardware implementation result

FPGA is a platform to implement hardware in the gate level abstraction. Since the designs are executed in parallel, speed of the FPGA implementation of any system is higher than speed of its software implementation. Furthermore, FPGAs are reconfigurable device, minimizing time-to-market, and simplifying verification and debugging [17]. So, FPGA is one of the best platforms

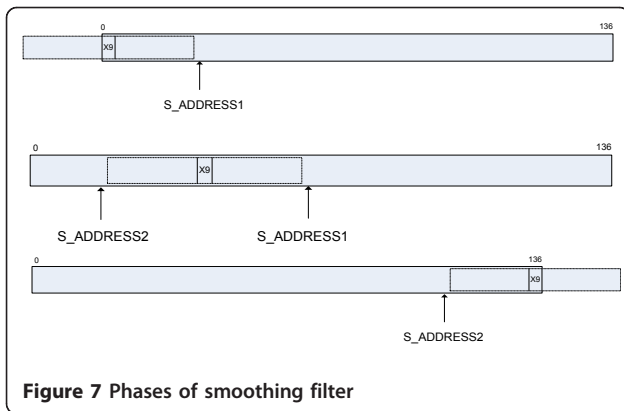


(a)



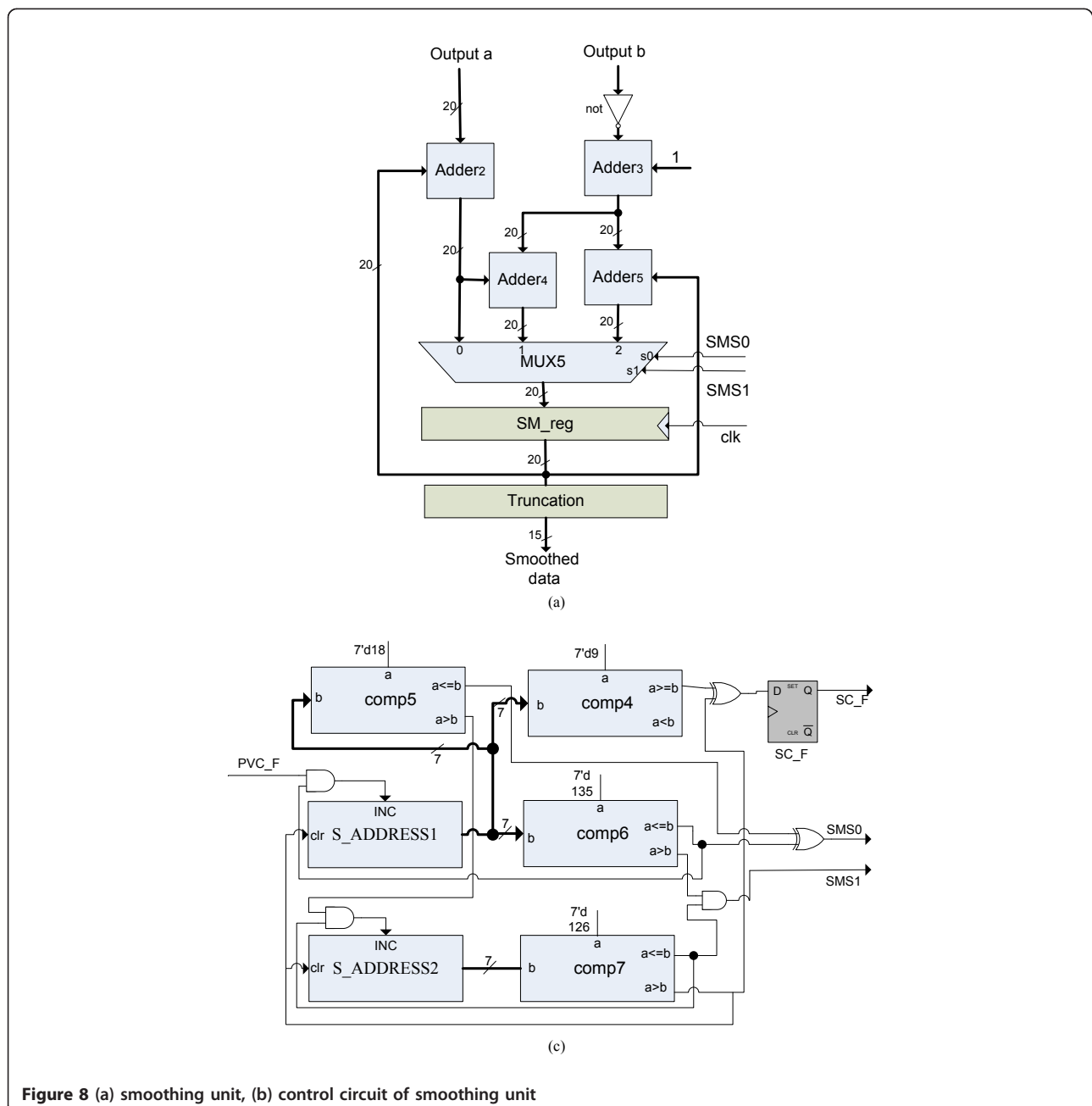
(b)

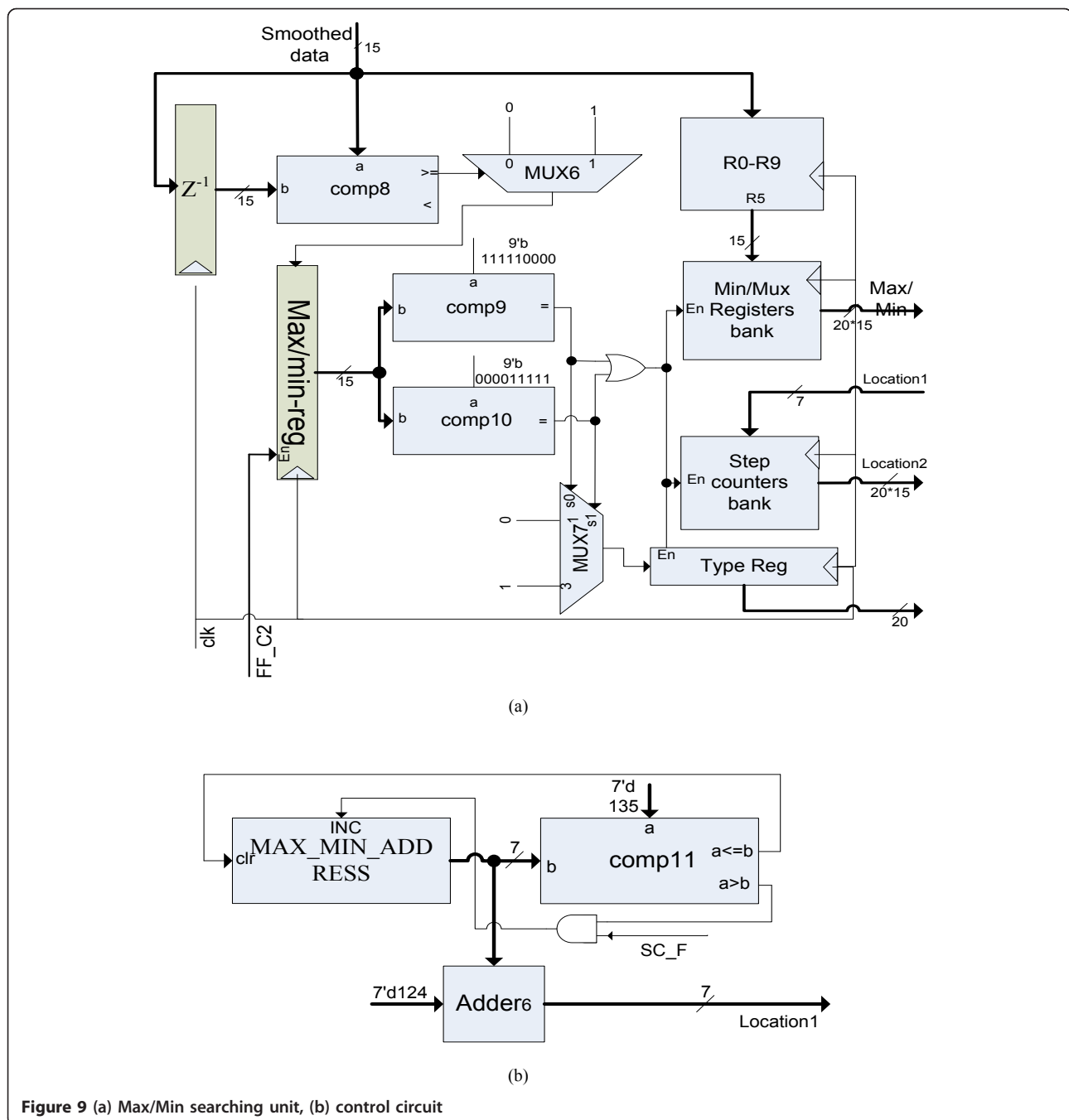
Figure 6 Projection vector circuit design. (a) PV_RAM and its ports, (b) control circuit to obtained



to implement a real time such as driver drowsiness system.

The proposed algorithm is implemented on an FPGA platform. In this implementation, the XC3SD3400A FPGA of Spartan 3-A DSP family, with many DSP slices, is used [26]. The resource requirements for proposed algorithm, including number of Slices, number of Slice Flip Flops, number of 4 input LUTs, and number of block RAM, are shown in Table 3. The maximum frequency obtained for this design is 83.1 MHz. To process an eye image of 136×82 pixels, the system needs 11310 clocks, for a total time of $136 \mu s$. This result shows that the proposed design,





and its hardware implementation, of an eye state recognition system is very suitable to work in a real-time system.

Comparison

Authors of [5] proposed a combination algorithm to detect the eye state, which reached 95% accuracy. This algorithm has five phases that each phase has many computations. Therefore, this algorithm has low speed and cannot be implemented on FPGA. In articles [6,7],

the authors obtained 89% accuracy. Their results are valid only under certain conditions; and changes in environmental light can affect the accuracy of these algorithms. Algorithm presented in [8] used the Hough transform to detect iris circle. This algorithm also is hard to implementation on FPGA. Although, the algorithm presented in [16] has high detection rate but it is complicated and hard to be implemented on hardware platforms. In Table 4, comparisons between different methods are shown.

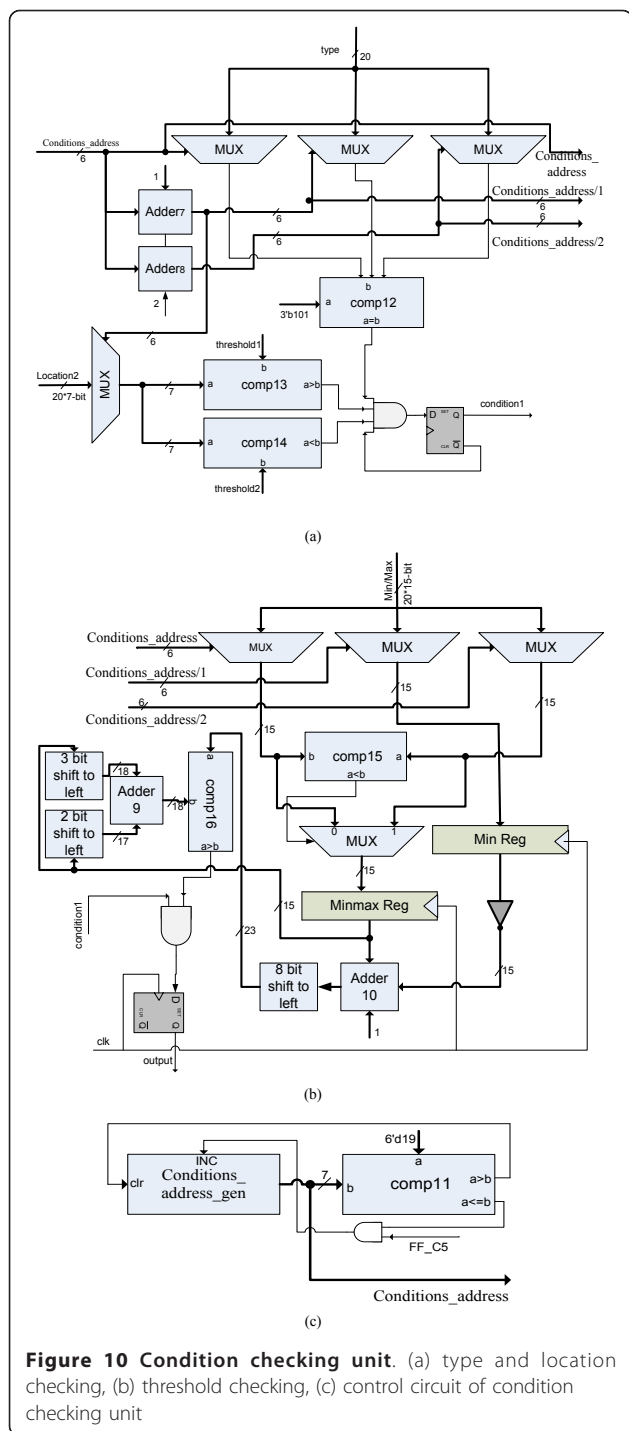


Figure 10 Condition checking unit. (a) type and location checking, (b) threshold checking, (c) control circuit of condition checking unit

Table 3 FPGA resource requirements to implementation proposed algorithm result

Logic utilization	Used	Available	Utilization (%)
Number of slices	1136	23872	4
Number of slice flip flops	836	47744	1
Number of 4 input LUTs	2071	47744	4
Number of BRAMs	1	126	0

Table 4 Comparison

	Database	Accuracy (%)	Speed	Hardware implementation
Our algorithm	The Caltech Frontal Face Database	89	Very high	Yes
[5]	Unknown	95	Very low	No
[6]	Unknown	89	High	No
[8]	Unknown	97	Very low	No
[16]	Unknown	98	Low	No

A computer simulation is developed to evaluate the proposed algorithm. The results show that a good balance between speed and accuracy obtained in our research, compared to other articles. An FPGA implementation also is presented to obtain a high speed. The FPGA implementation result shows that the proposed algorithm and its implementation method have a high speed and accuracy. Therefore, our proposed system can be used in real-time applications.

Conclusion

In this article, an algorithm to determine the state of an eye by using its image was presented. In our algorithm, we used the fact that the pupil and iris are darker than white part of the eye. We used the vertical projection to distinguish the state of the eye. The proposed method performed well in different light and eye color conditions. A computer simulation showed that the proposed algorithm has 89% accuracy.

Hardware design and implementation of the proposed algorithm were also presented. In this design, the algorithm is partitioned into five units and in each unit, the parallel and pipeline architectures were used. This implementation used 4% of the XC3SD3400A FPGA of Spartan 3-A DSP family. Result of this hardware implementation showed a fast architecture, i.e., the total time to process the image of an eye with 136 × 82 pixels in this system is 136 μs. This is a hardware implementation of an eye state recognition for real time.

Competing interests

The authors declare that they have no competing interests.

Received: 22 May 2011 Accepted: 15 February 2012
 Published: 15 February 2012

References

1. Drowsy Driving. http://dimartinilaw.com/motor_vehicle_accidents/car_accidents/drowsy_driving/. Accessed 2 May 2011
2. Newsinferno news site. <http://www.newsinferno.com/accident/drowsy-driving-behind-1-in-6-fatal-traffic-accidents>. Accessed 2 May 2011
3. P Boyraz, JHL Hansen, Active accident avoidance case study: integrating drowsiness monitoring system with lateral control and speed regulation in

- passenger vehicles, in *IEEE International Conference on Vehicular Electronics and Safety, ICVES 2008* 293–298 (2008)
4. K Yun Seong, L Haet Bit, K Jung Soo, B Hyun Jae, R Myung Suk, P Kwang Suk, ECG, EOG detection from helmet based system, in *6th International Special Topic Conference on Information Technology Applications in Biomedicine, 2007. ITAB 2007* 191–193 (2007)
 5. PR Tabrizi, RA Zoroofi, Drowsiness detection based on brightness and numeral features of eye image, in *Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2009. IIH-MSP'09* 1310–1313 (2009)
 6. WB Horng, CY Chen, Improved driver fatigue detection system based on eye tracking and dynamic template matching ICS. 11–2 (2008)
 7. H Wen-Bing, C Chih-Yuan, C Yi, F Chun-Hai, Driver fatigue detection based on eye tracking and dynamic template matching, in *Proceeding of the 2004 IEEE International Conference on Networking, Sensing & Control, Taipei, Taiwan, (21–23 March, 2004)*
 8. T D'Orazio, M Leo, C Guaragnella, A Distanto, A visual approach for driver inattention detection. *Pattern Recogn.* **40**, 2341–2355 (2007). doi:10.1016/j.patcog.2007.01.018
 9. Z Zhang, J-S Zhang, Driver fatigue detection based intelligent vehicle control, in *Paper presented at The 18th IEEE International Conference on Pattern Recognition (2006)*
 10. MS Devi, PR Bajaj, Driver fatigue detection based on eye tracking, in *First International Conference on Emerging Trends in Engineering and Technology* 649–652 (2008)
 11. Z Liu, H Ai, Automatic eye state recognition and closed-eye photo correction, in *19th International Conference on Pattern Recognition 1–4 (2008)*
 12. J Wu, T Chen, Development of a drowsiness warning system based on the fuzzy logic images analysis. *Expert Syst. Appl.* **34**(2), 1556–1561 (2008). doi:10.1016/j.eswa.2007.01.019
 13. Y Lei, M Yuan, X Song, X Liu, J Ouyang, Recognition of eye states in real time video, in *International Conference on Computer Engineering and Technology, ICCET'09* 554–559 (2009)
 14. T Hong, H Qin, Q Sun, An improved real time eye state identification system in driver drowsiness detection, in *IEEE International Conference on Control and Automation.* **0**, 1449–1453 (2007)
 15. Y-S Wu, T-W Lee, Q-Z Wu, H-S Liu, An eye state recognition method for drowsiness detection, in *IEEE 71st Vehicular Technology Conference 1–5 (2010)*
 16. MJ Flores, JM Armingol, A de la Escalera, Driver drowsiness warning system using visual information for both diurnal and nocturnal illumination conditions. *EURASIP J. Adv. Signal Process.* **2010**, 1–20 (2010)
 17. P Viola, M Jones, Rapid object detection using a boosted cascade of simple features, in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001.* **1**, 1-511–1-518 (2001)
 18. F Smach, M Atri, J Mitéran, M Abid, Design of a neural networks classifier for face detection. *Eng. Technol.* 124–127 (2005)
 19. C Huang, H Ai, Y Li, S Lao, High-performance rotation invariant multiview face detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **29**(4), 671–686 (2007)
 20. Z Jin, Z Lou, J Yang, Q Sun, Face detection using template matching and skin-color information. *Neurocomputing* 636–645 (2006)
 21. Y Jie, L Xufeng, Z Yitan, Z Zhonglong, A face detection and recognition system in color image series. *Math. Comput. Simulat.* **77**(5-6), 531–539 (2008). doi:10.1016/j.matcom.2007.11.020
 22. P Zhang, A video-based face detection and recognition system using cascade face verification modules, in *Proc of the 2008 37th IEEE Applied Imagery Pattern Recognition Workshop, Washington DC, 1–8 (2008)*
 23. V Vezhnevets, A Degtiareva, Robust and accurate eye contour extraction. *Proc. Graphicon-2003*, Moscow, Russia, 81–84 (September, 2003)
 24. Caltech vision lab. <http://www.vision.caltech.edu/html-files/archive.html>. Accessed 20 July 2011
 25. Eye SBU database. http://faculties.sbu.ac.ir/~eshghi/sbu-database-9008/one_eye.rar
 26. Xilinx co, <http://www.xilinx.com>. Accessed 2 May 2011

doi:10.1186/1687-6180-2012-30

Cite this article as: Dehnavi and Eshghi: Design and implementation of a real time and train less eye state recognition system. *EURASIP Journal on Advances in Signal Processing* 2012 **2012**:30.

Submit your manuscript to a SpringerOpen® journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com