Implementation System of Human Eye Tracking Algorithm Based on FPGA

Zhong Liu[1, 2], Xin'an Wang[1, *], Chengjun Sun[1] and Ken Lu[3]

Abstract: With the high-speed development of transportation industry, highway traffic safety has become a considerable problem. Meanwhile, with the development of embedded system and hardware chip, in recent years, human eye detection eye tracking and positioning technology have been more and more widely used in man machine interaction, security access control and visual detection.

In this paper, the high parallelism of FPGA was utilized to realize an elliptical approximate real-time human eye tracking system, which was achieved by the series register structure and random sample consensus (RANSAC), thus improving the speed of image processing without using external memory. Because eye images acquired by the camera often generate a lot of noises due to uneven light and dark background, the preprocessing technologies such as color conversion, image filtering, histogram modification and image sharpening were adopted. In terms of feature extraction of images, the eye tracking algorithm in this paper adopted seven-section rectangular eye tracking characteristic method, which increased a section between the mouth and the nose on the basis of the traditional six-section method, so its recognition accuracy is much higher. It is convenient for the realization of hardware parallel system in FPGA. Finally, aiming at the accuracy and real-time performance of the design system, a more comprehensive simulation test was carried out.

The human eye tracking system was verified on DE2-115 multimedia development platform, and the performance of VGA (resolution: 640×480) images of 8-bit grayscale was tested. The results showed that the detection speed of this system was about 47 frames per second under the condition that the detection rate of human face (front face, no inclination) was 93%, which reached the real-time detection level. Additionally, the accuracy of eye tracking based on FPGA system was more than 95%, and it has achieved ideal results in real-time performance and robustness.

Keywords: Human eye tracking, FPGA, real-time performance, preprocessing, elliptic approximation.

[1] Peking University, Shenzhen Graduate Schools, Shenzhen, 518000, China.

[2] School of Electronics Engineering and Computer Science, Peking Universitye, Beijing, 100871, China.

[3] iFutureLab Inc.955 Alma St., Suite B, Palo Alto, CA 94301, USA.

[*] Corresponding Author: Xin'an Wang. Email: keo201806@sina.com.

1 Introduction

Human eyes, which are responsible for acquiring most of human information, are an important sensory organ. In recent years, with the development of sensor technology and microelectronics technology, human eye tracking technology has been advanced by leaps and bounds, and it has been widely applied in the field of intelligent robot, human-computer interaction, automatic control and so on. Human eye tracking refers to the process of recognizing human eye activity that uses the camera to shoot human eye activity in real time and then adopts certain algorithm to transform the collecting human eye signal [Li, Winfield and Parkhurst (2005); Dohi, Yorita, Shibata et al. (2011)].

Due to the rapid development of the transportation industry, highway traffic safety has become a considerable problem. According to statistics, the traffic accident rate caused by drivers' fatigue driving reaches up to 40%, indicating that fatigue driving is one of the critical factors which influence traffic safety. Early fatigue detection methods are as follows: EEG tester is adopted to analyze the drivers' brain waves, thus judging the fatigue situation through corresponding brain waves [Fei, Wan, Wang et al. (2014)]; PERCOLS algorithm [Li, Xie and Dong (2011); Singh, Bhatia and Kaur (2011)] was improved by drivers' fatigue detection method based on human eye positioning method; Based on the statistical analysis results of Kolmogorov 2013 Smirnov Z test, the myoelectric peak factor and the maximum value of the myoelectricity and ECG correlation curve were selected as the comprehensive features to detect drivers' fatigue [Fu and Wang (2012)]; Based on EMG and ECG acquired by the portable real-time non-contact sensors, three physical characteristics (the complexity of electromyographic signal, the complexity and ample entropy) of the ECG signal were extracted and analyzed to detect drivers' fatigue [Wang, Wang and Jiang (2017)], and then on the basis of optical eye tracking system, infrared was used to replace general light source to closely irradiate the eyes. Because human eyes have no light perception to infrared light sources, there would be no rejection; The significant differences existing in the iris, pupil and sclera in human eyes for the refractive and reflective rate of infrared ray were utilized to locate iris and pupil [Zhu, Ji, Fujimura et al. (2002); Hutchinson (1989)], etc. However, most of these methods exceed or lag behind detection, so they don't meet the requirements for real-time detection. Moreover, these methods require special equipment for monitoring, so they are extremely inconvenient. Because of a great correlation between eye state and fatigue, it is an effective method to detect the drivers' fatigue state through detecting the blinking frequency of drivers' eyes and the response of the pupils to external stimuli in real time. Therefore, adopting parallel system FPGA to realize tracking algorithm and subsequent signal processing has great advantages and important significance [Theochrides, Link, Vijaykrishnan et al. (2004); Douxchamps and Campbell (2008)]. In this paper, the image of human face was acquired through the camera, and the image of human face was conducted with data preprocessing such as color conversion, image filtering, histogram modification, image sharpening and so on. After data preprocessing, the feature extraction of the eye tracking algorithm was carried out by using the seven-section rectangle tracking characteristic method. Meanwhile, the software algorithm system was implemented in the hardware parallel system FPGA. Finally, aiming at the accuracy and real-time performance of the design system, a more comprehensive

simulation test was carried out, and the results proved that the design of the system was practicable, and it could meet our design requirements.

2 Algorithm

2.1 Preprocessing

The eye image captured by the camera often generates a lot of noises due to uneven light and dark background. Fig. 1 presents the eye image and 3-D image in Matlab. The area of pupil and iris is basically corresponding to the low ebb in the image, but it can be seen from the three-dimensional diagram that there are multiple uneven low-ebb areas, which is caused by light and other noises. If the preprocessing of the eye image is not conducted, the results obtained will be deviated greatly. Therefore, the acquired eye object must be pretreated to eliminate the effects of light, etc.
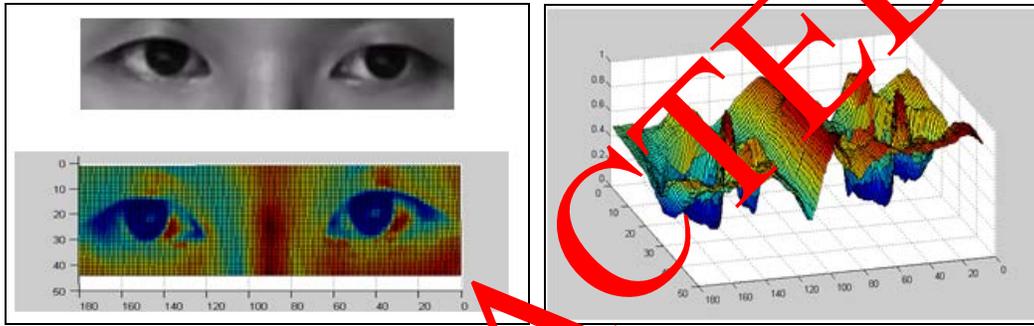

Figure 1: Matlab 3D image of eyes

The major preprocessing technologies include color conversion, image filtering, histogram modification, image sharpening, etc.

1. Color conversion

The image acquired by the camera is colorful, and it contains RGB three primary colors. Because color information is useless for detection, the image needs to be first converted into gray-scale map in order to avoid complex and a large amount of data calculation. Grayscale images reflect continuous brightness and intensity information without containing any color information, so they are useful for subsequent processing [Cui, Hu, Razdan et al. (2010)].

2. Image filtering

Image filtering refers to the operation that reduces or eliminates the noise of the original image but attempts to reserve the details contained in the image as far as possible. The nonlinear median filtering technology adopted in this paper aims to set the value of every bit of gray as the intermediate value of all pixel values within the pixel adjacency area. Therefore, this can eliminate light spots caused by light reflection [He, Sun and Tang (2010)].

3. Histogram modification

Histogram modification is a method adjusting the image histogram, which was put forward based on the idea of divided linear strength algorithm (Fig. 2). The main

practices are: first of all, calculate the pixel average of the image (Pa), and determine whether it is within the allowed range of the specified threshold (Th). If it is within this range, then Formula (1) is directly utilized to conduct a simple divided linear strength operation on the image. Otherwise, adjust the image pixel value to the specified average pixel value range, that is, adjust the image pixel value to the image pixel average in a specific environment according to Formula (2), in order to solve the problem of bright (or dark) image in the strong light (or weak light) environment. Then, Formula (1) is adopted to conduct divided linear strength operation, so as to strengthen image contrast and enhance image details.
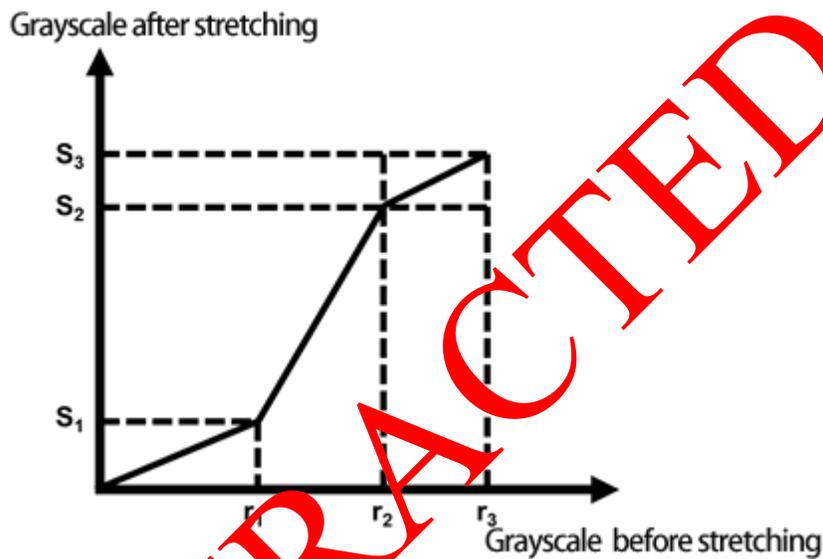

Figure 3: Schematic diagram of linear strength algorithm

Divided linear strength formula:

$$S = \begin{cases} \dfrac{s1*r}{r1} & 0 \leq r \leq r1 \\ \dfrac{(s2-s1)*(r-r1)}{r2-r1} + s1 & r1 \leq r \leq r2 \\ \dfrac{(255-s2)*(r-r2)}{r3-r2} + s2 & r2 \leq r \leq r3 \end{cases} \tag{1}$$

Formula of adjusting pixel value:

$$g(x, y) = \begin{cases} \dfrac{Th}{Pa}*f(x, y) & (\dfrac{Th}{Pa})*f(x, y) < 255 \\ 255 & (\dfrac{Th}{Pa})*f(x, y) \geq 255 \end{cases} \tag{2}$$

4. Image sharpening

In the process of acquiring, transmitting or processing image, a larger number of factors may make the image blurred, namely, image degradation. The essence of image blurring is caused by the average or integral operation. Therefore, the image can be conducted with inverse operation such as differential operation to make the image clear, namely, image sharpening. After sharpening, the contour of the eye pattern becomes clearer, which is more beneficial to detecting and localizing human eyes.

After the above preprocessing, a lot of noises in the image have been eliminated. Fig. 3 is the three-dimensional image of human eyes drawn in Matlab after preprocessing in Fig. 1. Compared with the eye image in Fig. 1, the 3D image after preprocessing has only two distinct low ebbs (corresponding to the two pupil zones). It is obvious that the noises have been basically filtered.
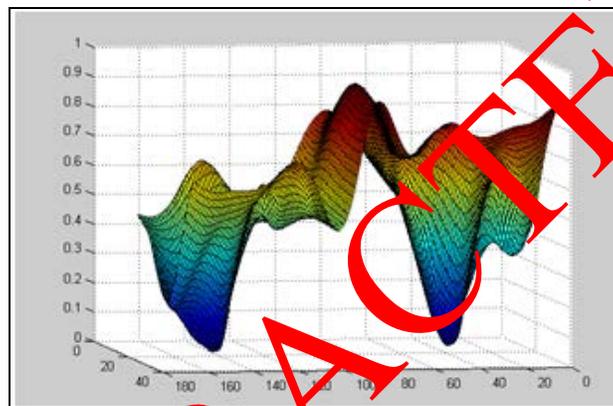

Figure 3: 3D image of human eyes after preprocessing

2.2 Feature extraction

Image features are the carrier of information, among which the edge is one of the most essential features of the image. On the one hand, the seven-section rectangle eye tracking characteristic method was adopted by the eye tracking algorithm in this thesis, and an added section between the mouth and the nose facilitated the identification of wrong information compared with the traditional six-section method. On the other hand, there are no complex recognition algorithms such as support vector machine (SVM) and neural network (NN) in the algorithm, so it is convenient for the realization of hardware parallel system in FPGA. The flow chart of eye tracking algorithm is shown in Fig 4. First of all, background detection is conducted to find the position of seven-section rectangle, and then the image integration is calculated in the selected area. Later, seven-section rectangular is utilized to scan the entire area to determine the area between the eyes. If eyes detected in this area, it indicates that viewers are watching the camera, otherwise, there are no viewers in specific area [Kawato, Testutani and Osaka (2005); Navabi (2007)].
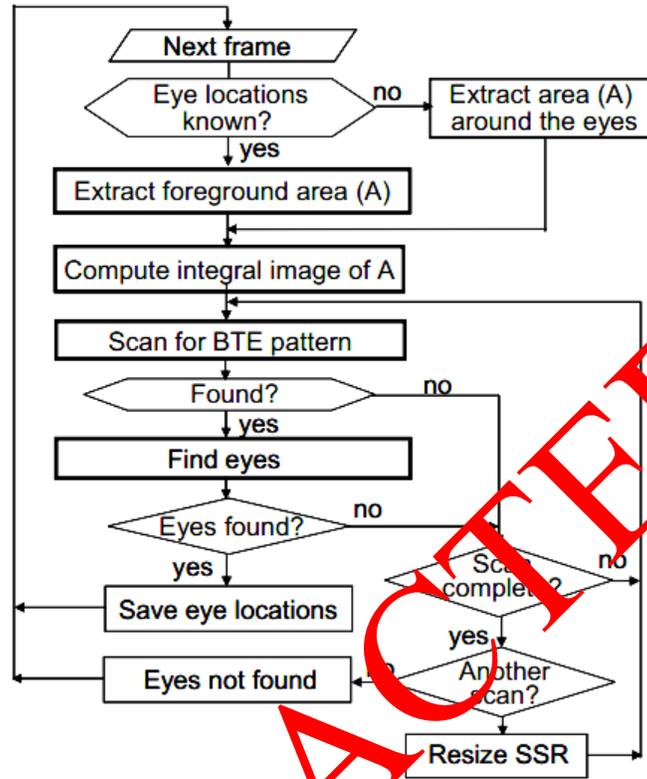
Figure 4: Flow chart of eye tracking algorithm

2.3 Elliptic approximation of feature points

Five feature points were randomly selected and utilized to produce an ellipse. The elliptic equation was assumed as below:

$$x^2 + Ax_i y + By_i^2 + Cx + Dy + E = 0$$

In the equation, the five parameters need to be determined by the feature points, so the synchronization equation of the elliptic approximation is:

$$\begin{pmatrix} \sum x_i^2 y_i^2 & \sum x_i y_i^3 & \sum x_i^2 y_i & \sum x_i y_i^2 & \sum x_i y_i \\ \sum x_i y_i^3 & \sum y_i^4 & \sum x_i y_i^2 & \sum y_i^3 & \sum y_i^2 \\ \sum x_i^2 y_i & \sum x_i y_i^2 & \sum x_i^2 & \sum x_i y_i & \sum x_i \\ \sum x_i y_i^2 & \sum y_i^3 & \sum x_i y_i & \sum y_i^2 & \sum y_i \\ \sum x_i y_i & \sum y_i^2 & \sum x_i & \sum y_i & \sum 1 \end{pmatrix} \begin{pmatrix} A \\ B \\ C \\ D \\ E \end{pmatrix} = \begin{pmatrix} -\sum x_i^3 y_i \\ -\sum x_i^2 y_i^2 \\ -\sum x_i^3 \\ -\sum x_i^2 y_i \\ -\sum x_i^2 \end{pmatrix}$$

Then, Cramer's Rule was utilized to solve the synchronization equation of the elliptic approximation, and this rule needs a large amount of calculation in the solution matrix. Assume the solution matrix is:

$$Ax = b$$

A is a n×n matrix; x and b are column vectors. On the basis of Cramer's Rule, it can be concluded:

$x_i = \dfrac{|A_i|}{|A|}$ The definition of Det(determinant) |A| is as follows:

$$|A| = \sum_{\sigma \in s_n} \text{sgn}(n) \prod_{i=1}^{n} A_{i,} \sigma_i$$

To realize the algorithm of multiple multiplication and addition operation in hardware usually needs to use ROM to store a part of operation symbols and constant. Assume n=5, the determinant |A| is expanded as follows:

$$\begin{aligned}
|A| = \ & a[1][1]*a[2][2]*a[3][3]*a[4][4]*a[5][5] \\
& - a[1][1]*a[2][2]*a[3][3]*a[4][5]*a[5][4] \\
& - a[1][1]*a[2][2]*a[3][4]*a[4][3]*a[5][5] \\
& + a[1][1]*a[2][2]*a[3][4]*a[4][5]*a[5][3] \\
& + a[1][1]*a[2][2]*a[3][5]*a[4][3]*a[5][4] \\
& \vdots
\end{aligned}$$

According to the rules, in the design, the operation symbols and row address were conducted with a tabular storage, and the multi-level pipeline architecture was used in the subsequent hardware implementation, which greatly improved the speed of calculating elliptic approximation and facilitated the calculation and acquisition of the maximum likelihood ellipse.

3 Hardware implementation

3.1 FPGA design overview

The system hardware block diagram is shown in Fig. 5. The pixel data obtained from the camera had a series of connection with preprocessing filter and then entered the data processing module. First of all, 128 feature points were detected, and invalid feature points were eliminated through finishing module. Later, consistency module was randomly sampled, and five points were collected from effective points, and they were stored in FIFO1. Elliptic module conducted floating arithmetic operation to determine the elliptic coefficient. Finally, the module was verified, and the number of inner models was calculated, and then the hypothesis was updated [Liu, Ma, Zhang et al. (2010)].
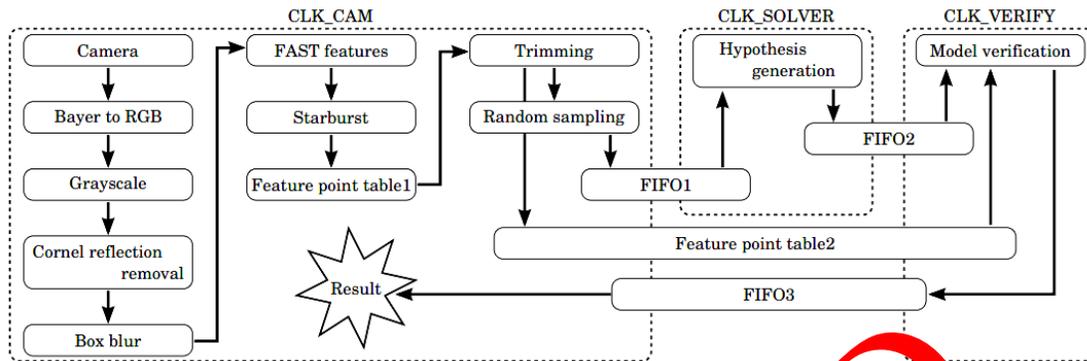
Figure 5: Block diagram of eye tracking system

3.2 Reprocessing

Series structure was utilized in the image preprocessing process, and this architecture was composed of effective region window register array, buffer pool, FIFOs and special operation pipeline operator. The whole preprocessing process can be directly completed in this architecture. At the same time, the entire preprocessing technology is mainly from color conversion to image filtering, then to histogram modification and image sharpening. In this way, all preprocessing modules can be realized in the series structure [Cho, Benson and Mirzaei (2009); Ngo, Tompkins, Feyzik et al. (2007); Lai, Savvides and Chen (2007)].

3.3 Pupil feature detection

The feature detection process is completed with 3 steps: (1) calculate the intensity gradient of all pixel points; (2) calculate the distance between pixel points and the center point as well as the coordinate angle; (3) update the table of feature points.

The intensity gradient adopted fast corner detection algorithm which is a feature extraction operator with high computational efficiency and high repeatability.

There was r* (r+1 /2) pixel points (p1, p2...p (r* (r +1 /2)) on a circle taking pixels as the center, whose fixed radius was r.

A threshold was defined. The pixel difference between the pixel points of the four corners and the central pixel point was calculated. If at least three of their absolute values exceeded the threshold value, they were regarded as the candidate corner, and then next investigation was conducted. Otherwise, they were not corner.

If p is candidate points, based on the classified corner feature detection, ID3 classifier was utilized to determine whether the candidate points had corner features according to 16 features, and the state of each feature was -1,0,1. Finally, non-maximal inhibition was adopted to verify corner feature [Laganière (2011); Cao and Zhang (2007)].

The multiplication, addition and inverse trigonometric functions provided by the hardware were used to calculate the distance between pixel points and center point, as well as the coordinate angle. A table containing 256 feature points was adopted to store the angle of the feature points and the distance between the feature points and the center

point. Once a new feature point was found to be closer to the center point, it would be updated. When all the pixels were scanned, all feature points about the algorithm architecture were stored in the table.

3.4 Random sample consensus

Random sample consensus module includes three clock domains, asynchronous FIFOs and feature point table, and it adopts dual-port RAMs to realize double buffering technology, and it is processed in parallel with the following several modules. The algorithm parallelization is realized through eliminating the iteration, and multiple "hypothesis-test" processes are conducted simultaneously, namely, several groups of samples are collected simultaneously in the data set, and then several models are estimated. The model validation module consists of synchronization equation, five feature points and feature solutions, and it realized the solution method of multiple real-time equation sets.

4 Results verification

Figures and tables should be inserted in the text of the manuscript.

4.1 Verification environment

DE2-115 development platform provided by Altera Company and its matched D5M camera were chosen for system verification, as shown in Fig. 6. The code development platform tool is Quartus II, which is a comprehensive PLD/FPGA development software, and the control code is written by Verilog HDL.
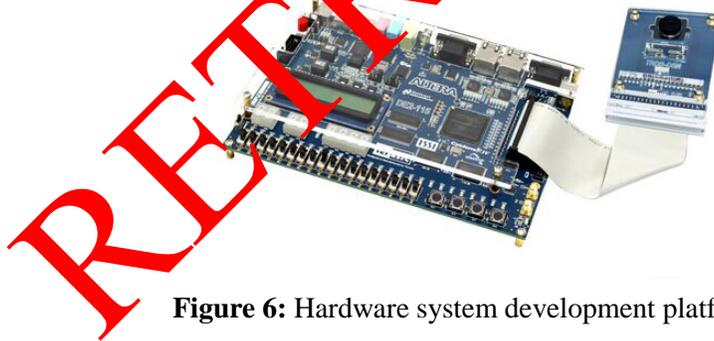

Figure 6: Hardware system development platform

4.2 Experimental results

In order to verify the tracking effect of human eye detection of the system designed in this paper, the human eye tracking experiment was conducted to a graph sequence actually photographed by the system. The algorithm was designed to meet the requirements of driving fatigue recognition, and it only considered the human eye tracking problem of rough front face, but it couldn't track the situation of side face. Moreover, the face sequence in fatigue detection can be summarized as the face sequence in front of the worktable, and its motion range is generally small. All the above problems

were considered when shooting the graph sequence, so the graphs which were taken were mostly front face with a small motion range. Therefore, this system was tested for many times, and its success rate is shown in Tab. 1:

Table 1: Number of frames of successfully tracking human eyes in experiment

	Experimental conditions	Number of frames successfully tracked	Success rate (%)
1	Front face, open eyes	93	89
2	Front face, close eyes	92	88
3	Face up	94	90
4	Face down	94	90
5	No face in image	105	100

Among these situations, only the observers in the first situation were staring straight ahead, which was considered to be a correct posture, and the observers in other situations did not look at the camera. Although the detection rate was different under different circumstances, the total tracking accuracy was relatively high (more than 88%).

When detecting fatigue driving, the detection speed is about 47 frames per second, and human beings' normal eye blinking time is about 0.2 s to 0.4 s. The data is 9 frames to 18 frames, and the probability of detection error in 9 consecutive frames of data is $5.16 \times 10^{(-9)}$, and the probability of detection error of 3 frames of data in the 9 frames is $1.728 \times 10^{(-3)}$. Meanwhile, seven of the nine frames of data are utilized to determine whether the eye blink is completely feasible. Therefore, it is thought that, when 3 frames of data have detection errors, the judgement of eye blink may be affected, and the probability that affects the judgement of eye blink is $1.728 \times 10^{(-3)}$. When six of the nine frames of data have detection errors, the probability that the judgment of eye blink is directly missed is $2.98 \times 10^{(-6)}$.
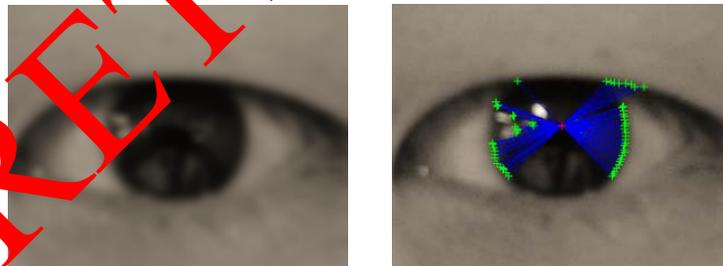

Figure 7: Effect drawing of tracking experiment

The results indicated that this system's detection speed was about 47 frames per second when the detection rate of human face (front face, no inclination) was 93%, which achieved the real-time detection level. However, the accuracy of eye tracking based on FPGA system was more than 95%, which has achieved ideal results in terms of both real-time performance and robustness. Meanwhile, the probability of errors in detecting human eye fatigue was less than 1%.

5 Conclusion

With the help of parallel processing capability of chip FPGA, this design aimed to solve the problem of limited processing capability and lagging detection results of the serial system in fatigue driving detection, so as to realize a system of real-time human eye tracking algorithm and enhance the parallel processing speed of image data. This design realized all algorithms in FPGA chip, including color conversion, image filtering, histogram modification, image sharpening, and random sampling elliptic approximation which solved the synchronization equation. It could be known from the verification experiment that the algorithm's computational results were good, and it realized a stable real-time processing system and had a little impact on the judgment of eye blink. The next work is to improve the tracking accuracy, optimize FPGA resource architecture, and add more mature random sampling algorithms, etc.

References

Cui, M.; Hu, J.; Razdan, A.; Wonka, P. (2010): Color-to-gray conversion using isomap. *Visual Computer*, vol. 26, no. 11, pp. 1349-1360.

Cho, J.; Benson, B.; Mirzaei, S.; Kastner, R. (2009): Parallelized architecture of multiple classifiers for face detection. *IEEE International Conference on Application-Specific Systems, Architectures and Processors*, pp. 75-82.

Douxchamps, D.; Campbell, N. (2007): Robust real time face tracking for the analysis of human behaviour. *International Conference on Machine Learning for Multimodal Interaction*, vol. 22, pp. 1-10.

Dohi, K.; Yorita, Y.; Shibata, Y.; Oguri, K. (2011): Pattern compression of FAST corner detection for efficient hardware implementation. *International Conference on Field Programmable Logic and Applications*, pp. 478-481.

Fei, W.; Wang, S.; Wang, X.; Peng, Y.; Yang, Y. D. (2014): Driving fatigue detection based on EEG recognition and vehicle handling characteristics. *Chinese Journal of Scientific Instrument*, vol. 35, no. 2, pp. 398-404.

Fu, R.; Wang, H. (2014): Detection of driving fatigue by using noncontact emg and ecg signals measurement system. *International Journal of Neural Systems*, vol. 24, no. 3.

Gao, F.; Zhang, X. M. (2007): Precise eye-corner locating in eye-gaze tracking system. *Computer Engineering,* vol. 33, no. 18, pp. 199-1200.

Hutchinson, T. E.; White, K. P. J.; Martin, W. N.; Reichert, K. C.; Frey, L. A. (1989): Human-computer interaction using eye-gaze input. *IEEE Transactions on Systems Man & Cybernetics*, vol. 19, no. 6, pp. 1527-1534.

He, K.; Sun, J.; Tang, X. (2013): Guided image filtering. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 35, no. 6, pp. 1397-1409.

Kawato, S.; Tetsutani, N.; Osaka, K. (2005): Scale-adaptive face detection and tracking in real time with SSR filters and support vector machine. *IEICE Transactions on Information and Systems*, no. 12, pp. 2857-2863.

Li, D.; Winfield, D.; Parkhurst, D. J. (2005): Starburst: A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches. *IEEE Vision for Human-Computer Interaction Workshop at CVPR*, pp. 1-8.

Li, L., Xie, M.; Dong, H. (2011): A method of driving fatigue detection based on eye location. *IEEE, International Conference on Communication Software and Networks*, pp. 480-484.

Liu, R.; Ma, S.; Zhang, M.; Wang, L. (2010): Implementation and optimization of the eye gaze tracking system based on DM642. *Third International Conference on Intelligent Networks and Intelligent Systems*, vol. 39, pp. 48-51.

Lai, H. C.; Savvides, M.; Chen, T. (2007): Proposed FPGA hardware architecture for high frame rate (≫100 fps) face detection using feature cascade classifiers. *IEEE International Conference on Biometrics: Theory, Applications, and Systems*, pp. 1-6.

Laganière, R. (2011): *OpenCV 2 Computer Vision Application Programming Cookbook*. Packt Publishing Ltd.

Ngo, H. T.; Tompkins, R. C.; Foytik, J.; Asari, V. K. (2007): An area efficient modular architecture for real- time detection of multiple faces in video stream. *International Conference on Information, Communications & Signal Processing*, pp. 1-5.

Navabi, Z. (2007): *Verilog Digital System Design.* Publishing House of Electronics Industry.

Singh, H.; Bhatia, J. S.; Kaur, J. (2011): Eye tracking based driver fatigue monitoring and warning system. *India International Conference on Power Electronics*, pp. 1-6.

Theochrides, T.; Link, G.; Vijaykrishnan, N.; Irwin, M. J. (2004): Embedded hardware face detection. *IEEE International Conference on VLSI Design*, pp. 133-138.

Wang, L.; Wang, H.; Jiang, X. (2017): A new method to detect driver fatigue based on EMG and ECG collected by portable non-contact sensors. *Promet-Traffic-Traffico,* vol. 29, no. 5, pp. 479.

Zhu, Z.; Ji, Q.; Fujimura, K.; Lee, K. (2002): Combining kalman filtering and mean shift for real time eye tracking under active IR illumination. *International Conference on Pattern Recognition Proceedings*, vol. 4, pp. 318-321.