IST 2009 - International Workshop on Imaging Systems and Techniques
Shenzhen, China
May 11-12, 2009

Haar-like Features Based Eye Detection Algorithm and Its Implementation on TI TMS320DM6446 Platform

Jiarui Zhou, Lai Jiang, Zhen Ji, Linlin Shen
School of Computer and Software Engineering
Shenzhen University, Shenzhen, China 518060
Email: 030bug@gmail.com, futurejl8026@sina.com, {jizhen, llshen}@szu.edu.cn

***Abstract*—A haar-like features based eye detection algorithm is introduced and implemented on TI TMS320DM6446 platform in this paper. With fast searching speed and high detection rate, the algorithm is packed into codec and put on the DSP-side for remote call. Application on the ARM-side captures face images from camera and outputs detection results to computer monitor. Experimental results demonstrated that the implemented eye detection system on DM6446 can search and locate eyes effectively with little chip resources required.**

I. Introduction

Eye detection is an important subject in pattern recognition and computer vision, and have been employed in many fields such as safe monitoring, video conferencing and judicial expertise *et al.*. In face recognition system the features can be extracted more effective with eye detection, and other facial organs can be located more precisely. These applications usually have portability requirements. Finding a high performance human eye detection algorithm, and implementing it on portable platform is of practical value.

Many algorithms have been proposed for eye detection. Dawtung Lin *et al.* [1] proposed an eye detection algorithm using face circle fitting and dark-pixel filtering. Huchuan Lu *et al.* [2] presented a novel detection algorithm based on rectangle features and pixel-pattern-based texture features. With high computational complexity, the algorithm can locate eyes precisely. Guangyuan Zhang *et al.* [3] provided a SVM based real-time eye detection algorithm. The Hu invariant moments of the focusing region are calculated and employed in SVM model development. In this paper a haar-like features based eye detection algorithm is introduced. With low computational complexity and high detection rate, the algorithm is fit to be implemented on chips.

TI TMS320DM6446 is a high performance digital multimedia processing chip provided by Texas Instruments Corporation. The chip contains a TMS320C64x+ core and an ARM926EJ core. The C64x+ DSP core can run at maximum 594MHz and processing digital signal efficiently. The ARM core with embedding Linux operation system provides a convenient and stable platform for on-chip application development. In this paper the haar-like features based eye detection algorithm is implemented on Spectrum digital DM6446 DVEVM (DaVinci Evaluation Module) board.

The implemented haar-like features based eye detection system is relevant to many applications. Detection algorithm based on haar-like features can be used in searching and locating other kinds of objects. The implementation of eye detection algorithm on DM6446 chip gives an example of porting and optimizing imaging algorithms on portable platform. By employing the eye detection algorithm in image preprocessing, the performance of face recognition system is improved. The implemented system can also be used in many other applications such as driver fatigue detection [4] and intelligent camera adjustment.

This paper is organized as follows. In Section II, the haar-like features based eye detection algorithm is formulated. In Section III, we illustrate the porting strategy of the eye detection algorithm on DM6446 platform. In Section IV, performance of the implemented algorithm is illustrated. Finally, the conclusion is given in Section V.

II. Haar-like features based eye detection algorithm

Haar-like features are digital image features used in object detection and recognition. Images of typical haar-like features are shown in Fig. 1.

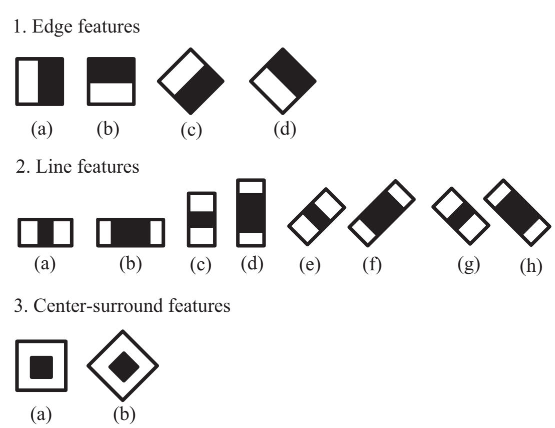
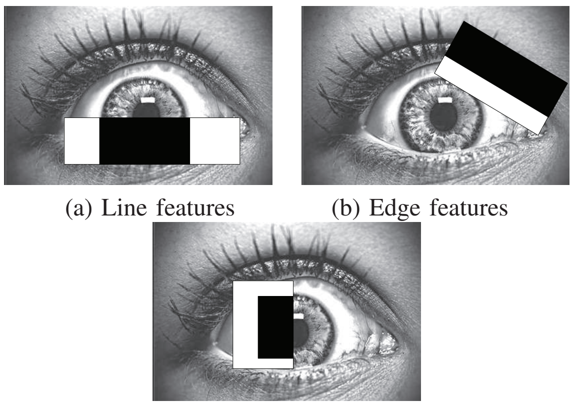


Fig. 1. Haar-like features.

Corresponding author: Prof. Z. Ji (Email: jizhen@szu.edu.cn).
This work was partially supported by the National Natural Science Foundation of China under grant no. 60872125 and 60572100, Royal Society (U.K.) International Joint Projects 2006/R3 - Cost Share with NSFC under grant no. 60711130233, and Science Foundation of Shenzhen City under grant no. 200706.

978-1-4244-3483-1/09/$25.00 ©2009 IEEE

Using original haar features in gray scale image object detection was firstly proposed by Paul Viola and Michael Jones [5]. Rainer Lienhart and Jochen Maydt [6] improved the algorithm by adding tilted haar-like features into the feature set, and used the extended set in face detection. The extended set of haar-like features is employed to detect eyes in gray scale images. The image of eye and three types of typical haar-like features are compared in Fig. 2. For the features are similar with the structure of human eye, the detection algorithm is effective.



(a) Line features (b) Edge features

(c) Center-surround features

Fig. 2. Comparison of haar-like features and eye image.

The eye detection algorithm employs AdaBoost cascade classifier working with haar-like features. The classifier is trained by giving positive samples (with eye image) and negative samples (without eye image) before detection. The typical distribution of an eye image on haar-like feature set is recorded in cascade classifier parameters. During detection, the captured gray scale image is split into image blocks and searched for eyes in sequence. The size of each image block is decided by a scan window. The distribution of each image block on the feature set is calculated and compared with the typical distribution of eye image. If most of the distribution is matched, the image block is considered to contain eye image, and vice versa.

For the position of facial organs are relatively constant, only reasonable region should be detected for eyes. After observing large number of samples can find that, by setting the top-left point of a face image as the origin of coordinate and normalized the height and width of the face to 1, the reasonable detection region of left eye is in the rectangle from point (0, 0) to (0.6, 0.6), and that of the right eye should be in the rectangle from point (0.4, 0) to (1, 0.6). The procedure of haar-like features based eye detection algorithm is shown in Table I.

TABLE I
PROCESS FOR EYE DETECTION ALGORITHM

Step 1	Initialize scan window size
	Split captured gray scale image into blocks
Step 2	Select next image block in sequence
	Performing eye detection algorithm on the image block
Step 3	If the image block contains eye image, go to step 6
Step 4	Go to step 2 until the whole image is detected
Step 5	Adjust the scan window size with scan window coefficient
	and go to step 1
Step 6	Output the results

III. EYE DETECTION SYSTEM ON DM6446 PLATFORM

TI TMS320DM6446 is a high performance digital multimedia processing chip. With ARM+DSP dual-core structure, the chip can process the control part and the computational part of an application separately, and makes the program run more efficiently. For the advantages of its powerful digital multimedia processing capacity and convenient developing environment, DM6446 is selected as the implementation platform for haar-like features based eye detection algorithm.

As shown in Fig. 3, the framework of DM6446 consists of two parts: the ARM-side part and the DSP-side part. An embedded operation system of MontaVista Linux runs on the ARM-side, and DSP/BIOS system runs on the DSP-side. These two sides communicate through the DSP-Link and shared memory space. On-chip applications run on the ARM-side with memory managed by Linux. Algorithms are packed into codecs according with xDM-xDAIS standard and handled to the Codec Engine. The codec can be put on the ARM-side and called by the application directly through the Codec Engine, or put on the DSP-side for remote call by using xDM Stub and Skeleton.

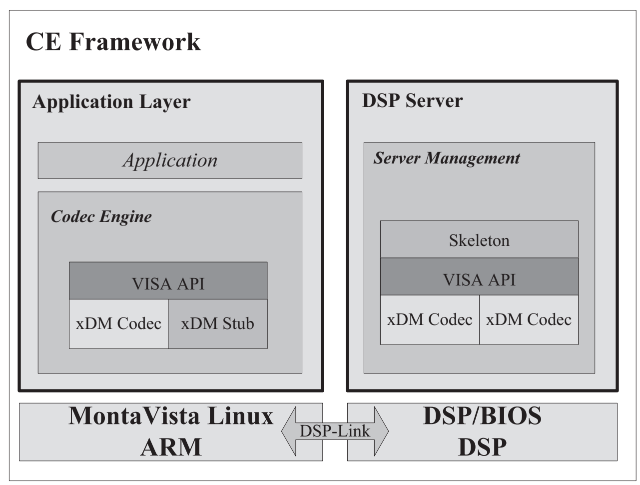


Fig. 3. DM6446 framework.

Haar-like features based eye detection system consists of three major parts: the capture part collects face images through a camera and transmits the image data to the application through the video process sub-system of DM6446 framework; eye detection algorithm codec is handled to the Codec Engine and called by the application for eyes searching and locating from the images; display part outputs the application interface and detection results on a computer monitor. The capture part and the display part are implemented on the ARM-side for their close relation with the application control. To satisfy the computation amount requirement of AdaBoost cascade classifier, the eye detection algorithm is packed into codec and run on the DSP-side for remote call. The system diagram is shown in Fig. 4.

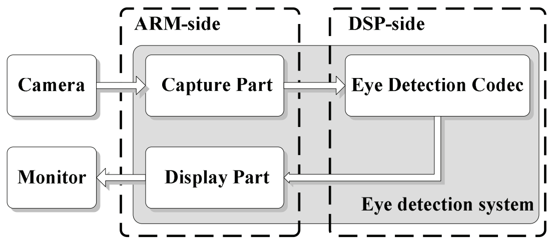


Fig. 4. System diagram of eye detection system.

The eye detection algorithm should be optimized for DM6446 platform. First, the program should be modified to adapt the peculiarity of C64x+ DSP core and improve its running speed. TI C62x/C64x FastRTS library is employed in program optimization. Table-driven approach is used for reducing pipeline broken. Experimental results show that the clock cycles and the computation time of implemented eye detection algorithm on DM6446 reduce remarkable after modification. Second, the parameters of AdaBoost cascade classifier are represented in floating-point numbers and the C64x+ is a fixed-point DSP core, the program should be converted to fixed-point. The scale factor Q should be determined for fixed-point conversion under two requirements: calculation after conversion should meet the precision requirements, and the operation results should be in the expression range of a fixed-point variable. After observing the running status of eye detection system can find that when setting Q as 24, the algorithm meets both two requirements and has the best detection performance. Third, the input parameters of eye detection system should be optimized for further improvement. Three major parameters are used in the algorithm: preprocessing scale coefficient, initial scan window size and scan window coefficient. By comparing the detection results on different parameter values can find that, when setting preprocessing scale coefficient as 6.4, initial scan window size as 24×24 pixels and scan window coefficient as 1.1, the algorithm can make optimum balance on detection rate and computation time.

In this paper we use Spectrum digital DM6446 DVEVM evaluation board as the implementation platform for the haar-like features based eye detection system. The evaluation board uses NTSC camera for image capture and output the result to computer monitor through an AV-VGA converter. The total memory space of the DVEVM evaluation board is 256MByte. By modifying the startup parameters and configuration files, 120MBytes of memory is allocated for Linux on ARM-side, 8MByte for ARM-DSP shared memory and 128MByte for DSP-side. The memory map of eye detection system on evaluation board is shown in Table II.

TABLE II
MEMORY MAP FOR EYE DETECTION SYSTEM

Side	Usage	Size
ARM	Linux memory	120MByte
ARM-DSP	Shared memory	8MByte
DSP	DSP heap	106MByte
	DDR (code, stack and static variables)	20MByte
	DSP-Link	1MByte
	Reset vector	1MByte

TABLE III
DETECTION RESULTS ON CMU-PIE DATABASE

Resolution	DR	DT	ARM load	DSP load
320×240	95.53%	0.13s	11%	24%
720×480	96.71%	0.19s	13%	28%

IV. EXPERIMENTAL RESULTS

In the first experiment the camera captured images are replaced by the CMU-PIE (Carnegie Mellon University Pose-Illumination-Expression) face image database [7] for eye detection. The CMU-PIE database contains about 40000 pictures of 68 peoples, including different pose, light condition and facial expression. The database is downloaded to the DM6446 evaluation board and searched for eyes with the haar-like features based eye detection system. The detection rate (DR), detection time (DT) and load of ARM core and DSP core on different image resolutions are collected as the experimental result of the implemented system. The results are shown in Table III.

The detection rates on both resolutions are higher than 95% and the detection time are less than 0.2s. This makes our implementation a real-time eye detection system. Meanwhile the ARM load and DSP load stay at a low level, so that the rest of the chip resource can be employed in other applications. The higher the resolution is, the higher the detection rate will be, and more detection time and core load is required.

In the second experiment the eye detection algorithm is employed in face recognition system to measure its boosting effects. Inaccurate face region determination and face angular deviation will reduce the recognition rate of a face recognition system. The angular deviation of the face image can be determined and corrected by calculating the angle between two eyes. By computing the distance between two eyes, the precise face region is found with typical human face proportion. By employing eye detection algorithm in face image preprocessing, the recognition rate of a face recognition system will be increased and the computational complexity will be reduced.

In this experiment the haar-like features based eye detection algorithm is used in the face recognition system based on Gabor wavelet [8] for face image preprocessing. The eye detection algorithm and the face recognition system will run on

TABLE IV
FACE RECOGNITION RESULTS ON BANCA DATABASE

System	RR	ER	RT
Without preprocessing	89.72%	13.18%	**2.39s**
With preprocessing	**95.22%**	**8.07%**	2.55s

TABLE V
DETECTION RESULTS ON ACTUAL RUNNING SYSTEM

Disturbance	Detection rate	Error rate
None	98.12%	0.15%
With glasses	95.33%	0.71%
One eye covered	87.92%	3.41%
Angular deviation	92.15%	1.26%

the DM6446 evaluation board. BANACA human face database [9] is employed in this experiment. By performing face recognition system with and without eye detection preprocessing on BANACA database, the recognition rate (RR), error rate (ER) and recognition time (RT) are collected as the experimental results and summarized in Table IV.

The recognition rate with eye detection preprocessing is much higher than that without preprocessing, and the error rate is lower. Meanwhile the increasing computation time caused by preprocessing is insignificant. The boosting effect of eye detection algorithm on recognition system is remarkable. The haar-like features based eye detection algorithm and its implementation on DM6446 platform can be used in other applications effectively.

In the third experiment the actual running system is employed in eye detection on real-time camera input images to illustrate its performance. Common interference cases such as wearing glasses, with angular deviation and with on eye covered are also added to evaluate the resisting disturbance capacity of the eye detection system. Each case is performed for 30 times. The detection rate and error detect rate are collected as the experimental results and shown in Table V.

The actual running system can detect eyes effectively even under the situation of having some common disturbances. Figure 5 shows the accomplished haar-like features based eye detection system on DM6446 evaluation board and Fig. 6 shows the actual running status of the implemented system with camera captured images as its input. The haar-like features based eye detection system on DM6446 is practical.

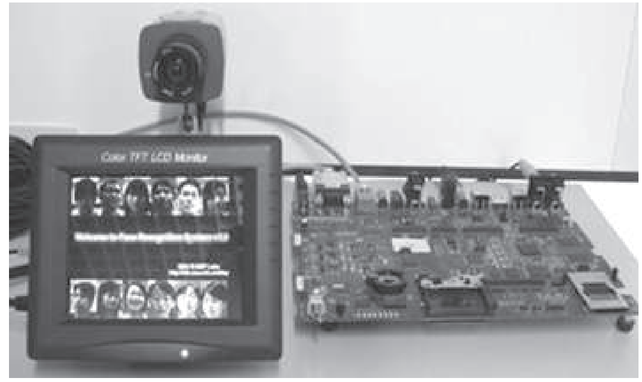
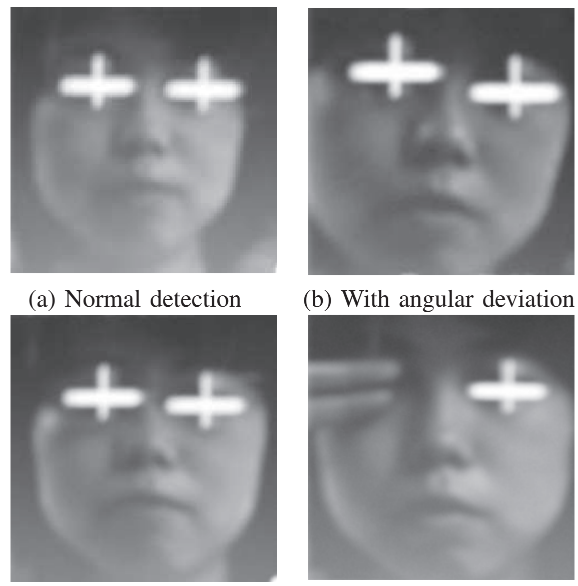


Fig. 5. Eye detection system on DM6446 evaluation board.



(a) Normal detection (b) With angular deviation

(c) With glasses (d) One eye

Fig. 6. Actual running eye detection system.

V. CONCLUSION

A haar-like features based eye detection algorithm is introduced in this paper and implemented on TI TMS320DM6446 platform. Experimental results demonstrated that the algorithm can search and locate eyes effectively with little chip resources. By employing the eye detection algorithm in face recognition system can increase recognition rate and reduce error rate remarkable. The implementation of haar-like features based eye detection algorithm on DM6446 platform is of practical value.

REFERENCES

[1] D.T. Lin and C.M. Yang, "Real-time eye detection using face circle fitting and dark-pixel filtering," in *Proc. of IEEE International Conference on Multimedia and Expo*, Taipei, Taiwan, 2004, pp. 1167-1170.

[2] H.C. Lu, W. Zhang, and D.L. Yang, "Eye detection based on rectangle features and pixel-pattern-based texture features," in *Proc. of International Symposium on Intelligent Signal Processing and Communication Systems*, Xiamen, China, 2007, pp. 746-749.

[3] G.Y. Zhang, B. Cheng, R.J. Feng, and J.W. Li, "Real-time driver eye detection method using support vector machine with Hu invariant moments," in *Proc. of International Conference on Machine Learning and Cybernetics*, Kunming, China, 2008, pp. 2999-3004.

[4] W.B. Horng, C.Y. Chen, Y. Chang, and C.H. Fan, "Driver fatigue detection based on eye tracking and dynamk, template matching," in *Proc. of IEEE International Conference on Networking, Sensing and Control*, Taipei, Taiwan, 2004, pp. 7-12.

[5] P. Viola and M. Jones, "Robust real-time object detection," in *Proc. of IEEE Workshop on Statistical and Computational Theories of Vision*, Vancouver, Canada, 2001, pp. 137-154.

[6] R. Lienhart and J. Maydt, "An extended set of haar-like features for rapid object detection," in *Proc. of IEEE International Conference on Image Processing*, New York, USA, 2002, pp. 900-903.

[7] T. Sim, S. Baker, and M. Bsat, "The CMU pose, illumination, and expression database," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, no. 12, pp. 1615-1618, December 2003.

[8] L.L. Shen and L. Bai, "MutualBoost learning for selecting Gabor features for face recognition," *Pattern Recognition Letters*, vol. 27, no. 15, pp. 1758-1767, October 2006.

[9] E. Bailly-Bailliere, S. Bengio, F. Bimbot, M. Hamouz, J. Kittler, J. Mariethoz, J. Matas, K.Messer, V. Popovici, F. Poree, B. Ruiz, and J.P. Thiran, "The BANCA database and evaluation protocol," in *Audio- and Video-Based Biometric Person Authentication* (pp. 625-638), German, Heidelberg: Springer Berlin, 2003.