Implementation of Iris Feature Encoding for Iris Recognition on TI DSP Board

S Sandeep Kumar
Department of Electrical Engineering
Indian Institute of Technology, Hyderabad
Hyderabad, India 502205
Email: ee13b1025@iith.ac.in

I. Introduction

Identification of persons is a very important need in the current age. Recognition of persons with their iris patterns i.e. Iris Recognition has been implemented and famous algorithms [1] for the same are being used all over the world. Iris recognition using this algorithm has been proven to provide no false matches in one-to-many searches even with databases of very large sizes. This has led to iris recognition become a standard biometric.

In this project, I have implemented the feature encoding module involved in iris recognition using a normalised iris image on the Texas Instruments TMS320C6670 [2] DSP Board. The details about how iris recognition is implemented is explained in section III.

II. Working Principle

The features of the iris are encoded into a bit steam like in Fig. 5 by a simple procedure (illustrated in Fig. 1) starting with the process of finding the iris in a given image. If the given image passes a certain focus level then it is used to find the iris and pupil circular curve fitting radius and center coordinates. This is mainly performed by maximizing an integro-differential operator [1].

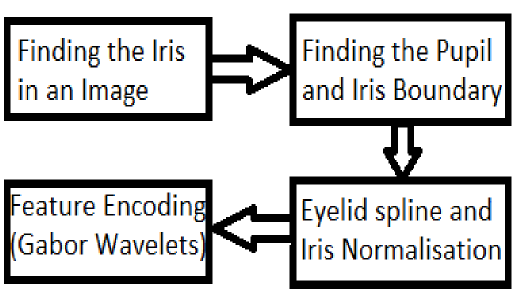

Fig. 1. Iris Feature Encoding Procedure

After this the eyelid boundaries are determined by fitting the spline parameters by statistical methods. If the given eye image has very little iris portion exposed it can't be used for feature encoding. In case the given eye image has sufficient portion of iris exposed then with the knowledge of the iris, pupil and eyelid boundaries, the portion of iris covered by the eyelid is marked off as noise. The output of this process on the image of an eye is shown in Fig. 4.

The remaining portion of the iris is normalised into a fixed size region to enable comparison of different irises. This process also corrects the issue of varying pupil dilation and difference in imaging conditions, detailed information on this is available in [1]. After the normalised iris image and the noise data are available, the feature encoding of the phase information contained in the iris is done using quadrature 2-D Gabor wavelets [1], [3]. The sign of the real and the imaginary parts obtained from 2-D Gabor wavelet demodulation is used to encode the bit stream. This encoded bit stream is then used to compare with other iris bit streams. The various algorithms used for matching different bit streams is detailed in [4].

III. Implementation

In this project I have implemented the feature encoding module of the iris recognition system. The normalised iris image and the noise data are taken as input from the open source MATLAB implementation of the complete iris recognition system of [4]. The generation of normalised iris data and mask data is implemented even in this open source MATLAB code as explained in Fig. 1. The individual modules are however implemented based on the techniques discussed in Section II as well as other techniques described in [4].

The feature encoding is performed by using 1D Log-Gabor filters. The input eye image from [5] is used to generate the required normalised iris image and noise data. The normalised iris image is convolved with the 1D Log-Gabor filter. Using the complex valued outputs of this process the iris biometric template and the noise mask are generated. The implementation of this project is available at [6].

The open source MATLAB implementation is input with an image from the sample iris images available at [5]. In this report a particular image shown in Fig. 2 is used to explain the procedure. The iris and pupil localisation operations and eyelid detection is then performed by this implementation. The output of this process is shown in Fig. 3. Using these parameters the normalised iris image and noise data are found. Essentially the region of the iris marked in Fig. 4 is normalised into a fixed size region.

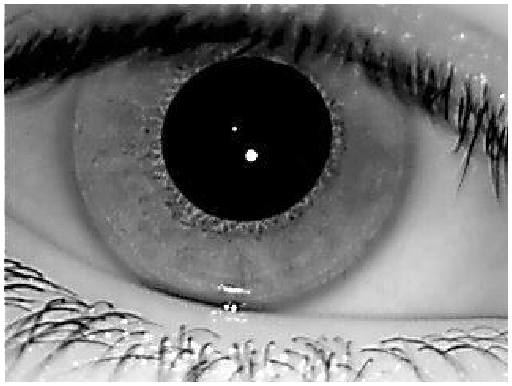

Fig. 2. Sample Eye Image from the IIT Delhi Iris Database

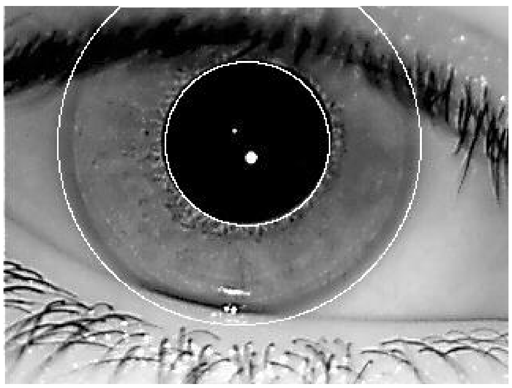

Fig. 3. Eye Image after Iris, Pupil localisation and Eyelid boundary detection

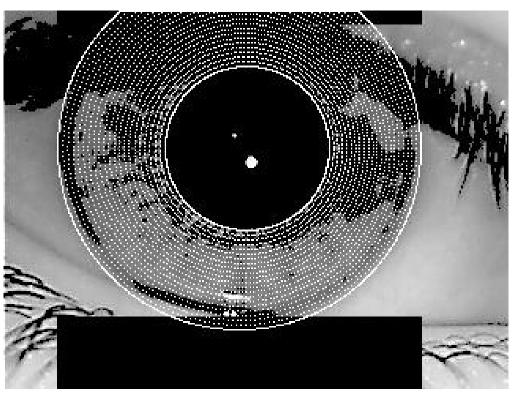

Fig. 4. Normalisation and Noise regions of the Iris in Fig. 2

IV. Results

The normalised iris image data and noise data output of the open source MATLAB implementation are converted to C definitions, to be passed to the code that runs on the Texas Instruments TMS320C6670 DSP Board, using a MATLAB script. The DSP board is controlled from Code Composer Studio [7] using an emulator on the DSP board. The implementation of the feature encoding from the normalised iris data stores the output into two integer arrays which can be accessed from the Memory Browser on Code Composer Studio. This data is then imported into MATLAB and is compared with the output of the open source MATLAB feature encoding implementation.

Fig. 5. Biometric Template for Iris in Fig. 2

Fig. 6. Noise Mask for Iris in Fig. 2

The biometric template for the iris in Fig. 2 is shown in Fig. 5. The noise mask obtained is shown in Fig. 6. The results were compared with the results obtained from the open source MATLAB implementation and were exactly matching for all the available sample images.

V. Conclusions and Future Directions

In this project, I have implemented the feature encoding module of a typical iris recognition system using the normalised iris image and noise data obtained from the open source MATLAB code of [4] and shown that they give exactly the same results.

I have made all my work free and open source except for the FFT [8] function that I have used which is available for non-commercial use only. This needs to be changed to make my work completely free and open source. Implementation of the remaining modules including the iris code matching block are required to make a complete iris recognition system possible on the DSP Board.

VI. Acknowledgement

I have used the sample iris images from IIT Delhi Iris Database [5] for which I am thankful to its owners and as a requirement have referred [9]. I am grateful to Dama Sreekanth, graduate student at IIT Hyderabad for his valuable guidance on using the DSP Board.

References

[1] J. Daugman, "How iris recognition works," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 14, no. 1, pp. 21–30, Jan 2004.

[2] "TMS320C6670 Multicore Fixed and Floating Point SoC," http://www.ti.com/product/tms320c6670.

[3] J. Daugman, "Complete discrete 2-d gabor transforms by neural networks for image analysis and compression," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 36, no. 7, pp. 1169–1179, Jul 1988.

[4] L. Masek, "Recognition of human iris patterns for biometric identification," Master's thesis, University of Western Australia, 2003. [Online]. Available: http://www.csse.uwa.edu.au/~pk/studentprojects/libor/

[5] "IIT Delhi Iris Database," http://www4.comp.polyu.edu.hk/~csajaykr/IITD/Database_Iris.htm.

[6] "Implementation of Iris Feature Encoding performed in this Project," Available: http://www.iith.ac.in/.

[7] "Code Composer Studio," http://www.ti.com/tool/CCSTUDIO.

[8] "A very fast Arbitrary-N Point FFT," http://www.corix.dk/html/fft.html.

[9] A. Kumar and A. Passi, "Comparison and combination of iris matchers for reliable personal authentication," *Pattern Recognition*, vol. 43, no. 3, pp. 1016 – 1026, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320309003343