5-2019

Design of an embedded iris recognition system for use with a multi-factor authentication system.

Christopher David Hess

Design of an embedded iris recognition system for use with a multi-factor authentication system.

An undergraduate Honors thesis submitted in partial fulfilment

of the requirements for the degree of

Bachelor of Science in Electrical Engineering

by

Christopher Hess

May 2019

University of Arkansas

Abstract

This paper describes in detail the design, manufacturing and testing of an embedded iris scanner for use with a multifactor authentication system. The design process for this project included hardware design from part selection to board design to populating. Additionally, this process included the entirety of the software development, though the iris recognition process was largely based on other works. The functional requirements for the overall multi-factor authentication system were to have three authentication methods with a thirty second window to complete all three. The system acceptance accuracy was required to be greater than 75%. Those requirements therefore dictate that the iris scanner module must also have an acceptance accuracy higher than 75% and perform iris recognition in a few seconds so that the user can gain admittance in the allotted window of time. While the hardware has been verified and tested, further development and testing is necessary on the software and image processing. This work is funded by the Department of Energy's Kansas City National Security Campus, operated by Honeywell Federal Manufacturing & Technologies, LLC under contract number DE-NA0002839.

Acknowledgements

There are several individuals who made this project possible. First, the team at Honeywell Federal Manufacturing & Technologies who commissioned and funded the Multi-Factor Authentication project. I would like to especially thank Rene Rodriguez and Doug Lee for their mentorship and reviewing our designs. Additionally, I would like to thank Robert Saunders, Ian Sanders and Cecily Redman for giving me direction for my design, helping me troubleshoot issues and for everything else they do for the senior design class. I would like to thank Dr. John Gauch for giving me ideas for iris recognition at the beginning of this project and for helping me with implementing it when it was crunch time. Portions of the research in this paper use the CASIA-IrisV4 collected by the Chinese Academy of Sciences' Institute of Automation (CASIA).

Table of Contents

List of Figures

Introduction

There is a distinct need for increased security measures for many applications. Traditional methods of security such as simple pins or passwords are becoming increasingly vulnerable and insecure. Possession based security factors, such as keys, have always been susceptible to malicious theft. An increasingly popular solution to security is to use biometrics and multifactor authentication. Biometric factors are generally not susceptible to theft and are not as easily compromised as a pin. The detrimental shortcomings of any one authentication technique can be minimized by combining multiple authentication factors. For example, using a simple password and thumbprint scanner together makes it unlikely that an imposter will gain access even if they have acquired a user's password. In this case an unauthorized access from a thumbprint scanner false positive is mitigated by requiring a password.

Honeywell Federal Manufacturing and Technology commissioned a project to create a multifactor authentication system with three authentication methods including an RFID card scanner and two biometrics. The system will operation requires authentication of all three authentication factors within 30 seconds, to unlock a magnetically latched door. If authentication fails three consecutive times, then the user will be locked out for 5 minutes.

Along with a RFID card scanner, a capacitive thumbprint scanner and iris scanner were selected to be used for the system. The scope of this paper from this point on is focused on the development of the embedded iris scanner development for this project. The decision to develop the iris scanner was undertaken because there was not an affordable, readily available product on the market that could directly interface with an embedded processor. Many iris scanners on the market are only image acquisition units with software drivers intended to be ran on a PC. While there are self-contained products that do on board image processing, they are not intended to be interfaced with embedded processors.

Background

Iris recognition has proven to be a very effective method of authentication. The human iris, similar to a fingerprint, is extremely unique. Unlike a finger or thumb print, an iris changes little throughout a person's lifetime [1] and the iris is not easy to modify or damage. As a result iris scanners do not face the same issues as fingerprint scanners do if a finger is dirty, scraped, etc. For these reasons, iris scanners are an excellent biometric identifier. Daugman [1] developed an algorithm to encode an iris to a 256-byte variable that can accurately be matched to the same iris with low false acceptance and low false rejections. The original Daugman algorithm created in 1993 estimated 1 in 151,000 false acceptances and 1in 128,000 false rejections [1]. This algorithm is still the most commonly used, although, it has been improved upon in recent years [8].

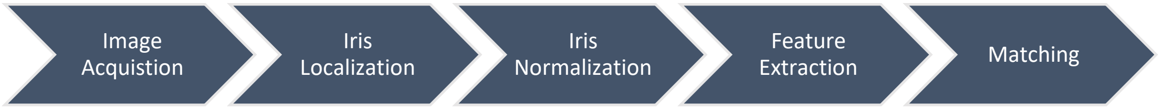

Figure 1. Five steps for iris recognition process

There are four high level steps for iris recognition, these are image acquisition, iris localization, iris normalization, feature extraction and matching. For image acquisition, an infrared picture of an eye should be taken while illuminated with Infrared (IR) light. Observing the eye under these conditions allows the texture of darker eyes to be more easily observed using low intensity IR light without causing the user discomfort that would be caused from visible light.

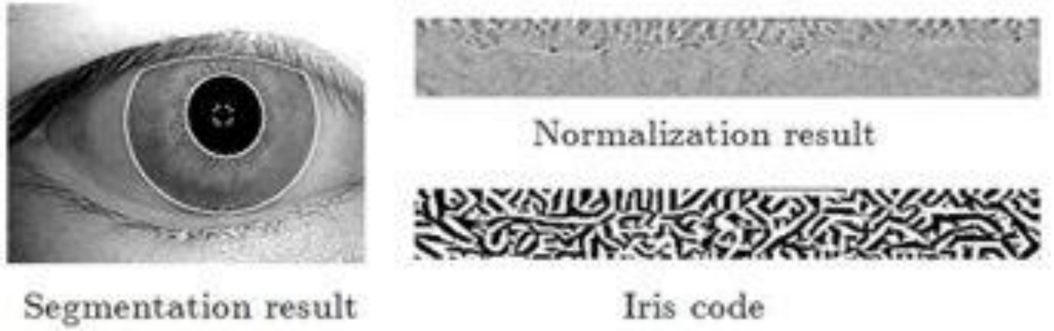

Figure 2 Sample from [2] of iris localization, normalization and feature extraction

Once an image is captured, the iris must be isolated from the rest of the image. The pupil and sclera are not of interest, so they must be removed from the image before further processing. A centroid method is used [2] to identify the center of the pupil and radius of the pupil. Identifying the outside (limbus) radius of the iris is more complex, [1] uses integrodifferential operators to search an image for circular edges. This operator from [1] is shown in (1).

$$max_{(r,x0,y0)} \left| G_{\sigma}(r) * \frac{\partial}{\partial r} \oint_{r,x0,y0} \frac{I(x,y)}{2\pi r} ds \right| \tag{1}$$

Next, polar coordinates are applied to the image of the iris. This allows the ring-shaped image to be saved in a rectangular form. Using polar coordinates allows the same regions of the iris to be referenced relative to the pupil and limbus boundaries [1]. As the iris contracts and expands in response to light, it is important to have a reference system that can identify patterns when the iris is stretched or compressed.

Perhaps the most important step is the feature extraction, where typically a two-dimensional Gabor filter is used to analyze the texture and patterns of the iris. Daugman presented the filter described in (2) in [1]. A similar filter is proposed in [2] and is shown in (3), where *u'* and *v'* are described in (4) and (5) respectively. By convolving either filter with a normalized image, an image can be encoded to a desired resolution. The original Daugman algorithm saved the encoded image in a 256-byte iris code [1].

$$G(r,\theta) = e^{-i\omega(\theta-\theta_0)} e^{-\frac{(r-r_0)^2}{\alpha^2}} e^{-\frac{(\theta-\theta_0)^2}{\beta^2}} \tag{2}$$

$$G(u,v,\theta,f,\sigma_u,\sigma_v) = \frac{1}{2\pi uv} e^{-\pi\left(\frac{u'^2}{\sigma_u^2}+\frac{v'^2}{\sigma_v^2}\right)} e^{i2\pi f u'^2} \tag{3}$$

$$u' = ucos\theta + vsin\theta \tag{4}$$

$$v' = -usin\theta + vcos\theta \tag{5}$$

Once an image has been fully processed and encoded, it can then be compared to other iris codes. An effective and simple method that has been used is to find the hamming distance between the two iris codes. Hamming distance is described as HD in (6) where A_i and B_i index the individual bits of two iris codes. It is unlikely that two images of the same iris will have the exact same iris code, but a threshold of typically 10% to 20% variance can be set as acceptable as generally a different iris will not match that closely [1].

$$HD = \frac{1}{N}\sum_{i=1}^{N} A_i \otimes B_i \tag{6}$$

Figure 2 shows sample hamming distances between two iris codes of the same eye (authentics) on the left and hamming distances between two iris codes of different eyes (imposters) on the right. The mean hamming distances for authentics is around 10% whereas the mean for imposters is closer to 50%. With this data a threshold could be set closer to 30% with neglible false acceptances and false rejections.

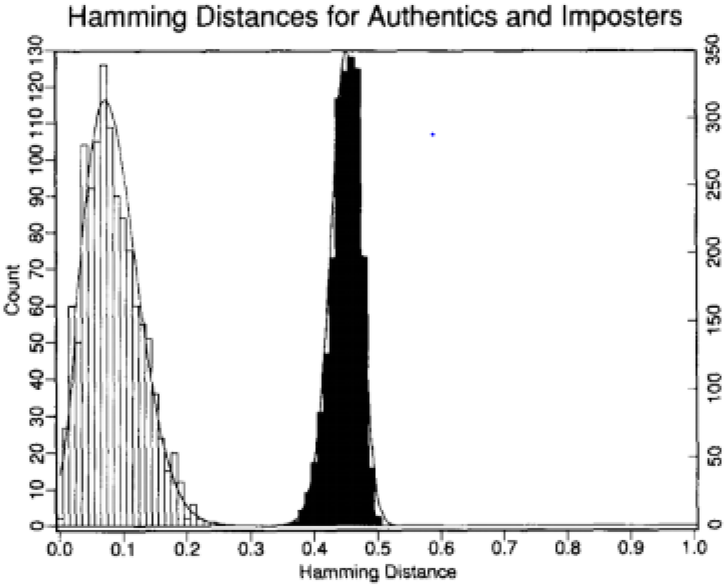

Figure 3. Hamming Distances for Authentics and Imposters

Hardware Considerations

Some research has been done specifically implementing iris recognition in an embedded system [2,3]. A TI DM6446 dual core chip was used in [2], where the 600 MHz DSP core was used for image processing. Alternatively, a blackfin DSP was used in [3].

Design

Hardware Selection and Desgin

The high level hardware design for this project is shown below in Figure 3. For the system to be used, the main board must send a command to match an iris and send an ID. The system will use this ID to load the users iris code to be compared. Next, the user must place their eye in position and then press a button. Data is captured from the IR camera and transmitted to a complex programmable logic device (CPLD), and from the CPLD to a PIC microcontroller. The CPLD is used to slow down data transmission and will be descirbed in detail later. Once the image has been fully transmitted, the PIC processes the image into a specified sized iris code. The iris code is loaded from memory and iris code of the new scan are compared, if they match then the PIC indicates to the main board that the scan was a success.

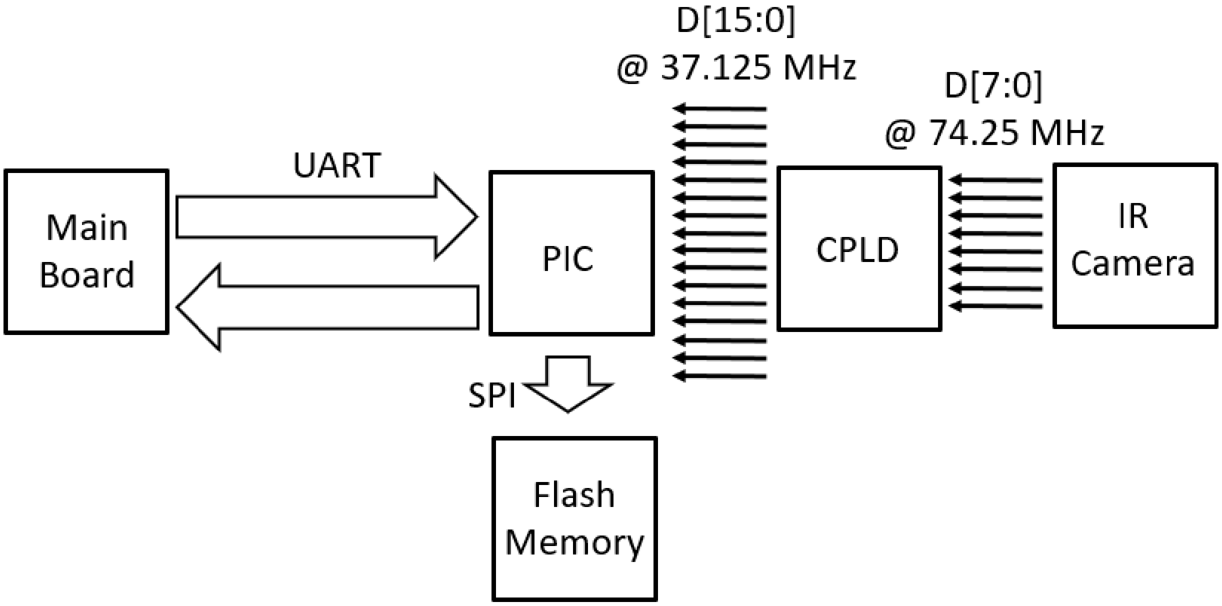

Figure 4. High level hardware design

The hardware design for this project was based largely based around a 1.3 MP, near infrared (NIR) camera. The camera selected was the e-CAM10_CU130_MOD. This camera was selected because it had good resolution, is sold with an appropriate lens for this application and had a relatively simple interface for an embedded project. An I2C communication interface is used for the PIC to send commands to the camera to capture an image. The camera uses a 12-bit parallel data interface to send the image as grayscale.

A PIC32MZ was selected to use with the NIR camera for several reasons. This chip is part of Microchip's graphics processing family, so it boasts features that are useful for image processing [4]. The first of these is 32 MB of on chip DDR2 DRAM. This is plenty of RAM to accommodate the 1.3 MB image being processed. With a 200 MHz clock and plenty of IO ports, this was the best chip available for interfacing with the selected camera that was not a Ball Grid Array (BGA) package (while BGA processors were considered, ultimately an exposed lead package was selected for manufacturing purposes).

For the initial design, the parallel data output from the camera was directly wired to the parallel master port (PMP) on the PIC. Upon further review, the PMP was incompatible with the camera due to configuration issues as well as timing. A CPLD was added to the design to reformat the data coming out of the camera effectively slowing the data rate. For this redesign, the 4 least significant bits of the camera output are ignored. Previous research used 8-bit grayscale and had satisfactory results [1,2,3], so this is not an issue. The most significant 8-bits of data are fed into the CPLD, along with the control signals from the camera and PIC. The data clock from the camera is reduced by half using a bit counter. This reduced clock is then used to latch data into two 8-bit registers; one positively edge triggered, and one negative edge triggered. A 16-bit register is used to concatenate the data from both registers on positive clock edges. Data is output from this register. This approach allows all data to be retained while reducing the data clock by half.

Figure 4 shows the full design of the operating logic for the CPLD. Signals f_valid and l_valid are control signals generated from the NIR camera while the capture signal is generated by the pic. The signal int tells the PIC when data is ready to be read. This signal is necessary to use the direct memory access (DMA) protocol on the PIC to read data from the CPLD as quickly as possible. This design was used to determine that 33 logic elements were needed, so an Altera Max V with 80 logic elements and plenty of IO pins was selected for this design.

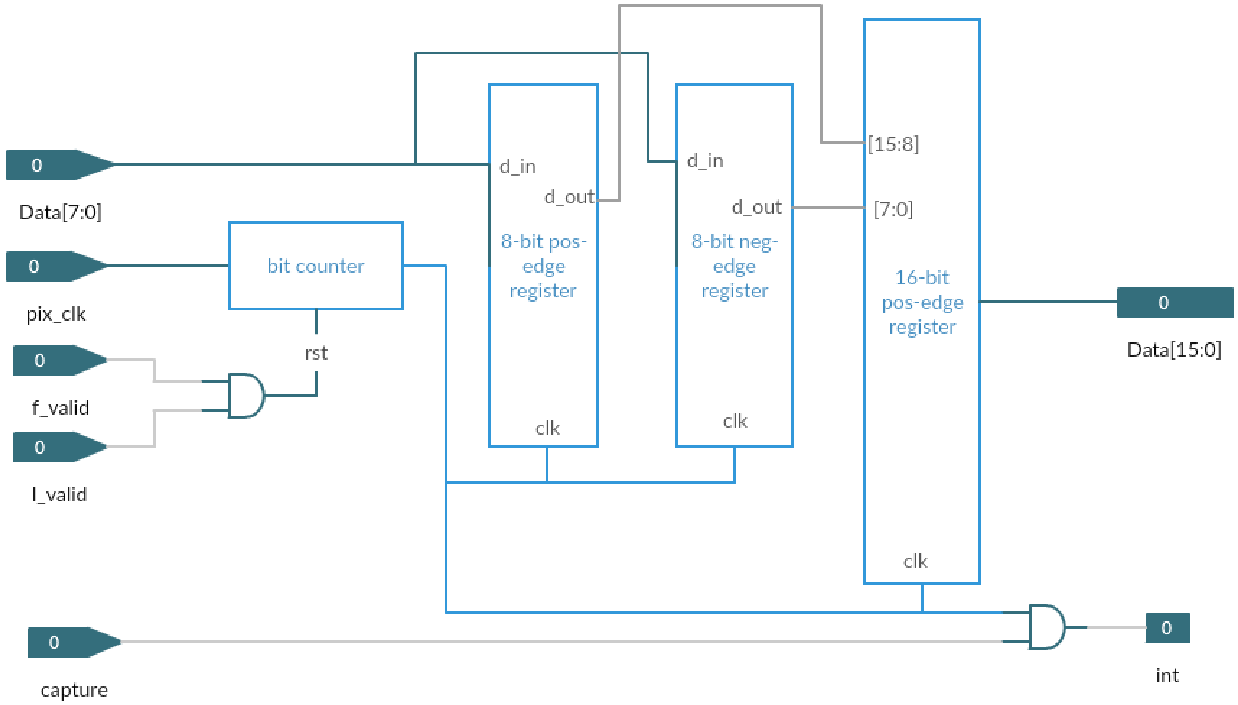

Figure 5 Logic Design for CPLD

The remaining components for the design are there to optimally run and interface the aforementioned devices. A half amp rated linear regulator is used to take in 3.3V from the main board and output 2.8V and 1.8V. These voltages were required for the NIR camera and therefore the remaining components were also spec'd to run on those voltages. Additionally, a 24 MHz crystal oscillator package was placed on the board to get a high-quality clock signal that is sufficient for high speed communications.

Board Design

A four-layer printed circuit board (PCB) was designed for this application. All parts where placed on the top layer of the board, shown below in Figure 5, so that the whole circuit could be easily populated and soldered. Four layers were used to accommodate the dual voltage levels and many signal traces from the camera data. The top and bottom layers were used for signals and the 1.8V traces while, the internal layers were planes for 2.8V and ground. Having an internal voltage and ground planes reduces interference between signals on the top and bottom layers while providing low impedance path for the power.

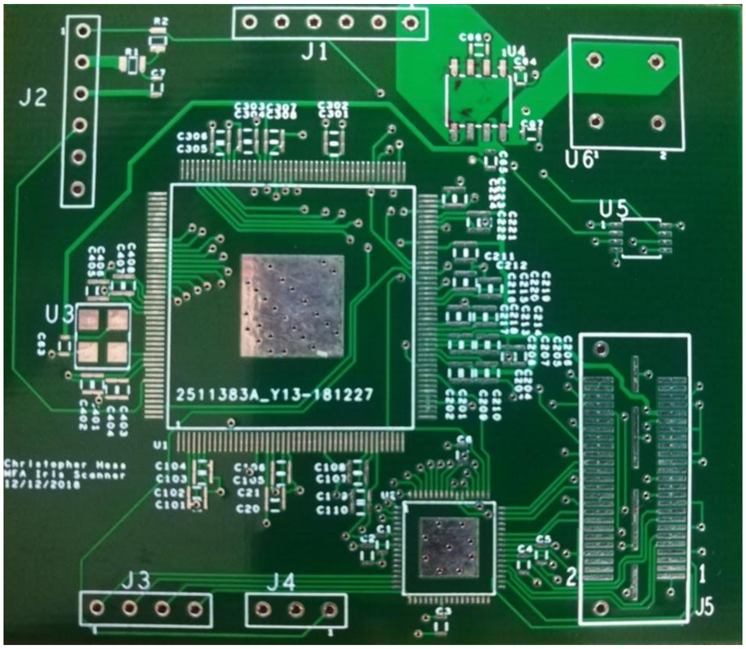

Figure 6 Unpopulated PCB for Iris Scanner

Primary PCB design concerns were minimizing high speed trace lengths, minimizing coupling capacitor trace lengths and maximizing trace width for power levels. The CPLD was placed as close as possible to the camera interface to minimize the 74.25 MHz data trace lengths, these components are seen in the bottom right of figure 5. The oscillator and CPLD were also placed as close as possible to the corresponding input pins on the PIC.

Image Processing

The first step for the image processing side is iris localization, this process is requires finding both the pupil and limbic boundaries of the iris. The pupil boundary tends to be better defined than the limbic so it easier to identify this first and use it to estimate the center of the iris. In this context Purkinje [2] spots are defined as the spots on the image of the iris caused by IR LEDs during image acquisition. Purkinje spot analysis is used in [2] to determine the quality of a collected images, and a similar analysis can actually be used to find the pupil boundary. First a binary image is created from the original grayscale image. A threshold is used so that essentially only the pupil remains. Then connected domain analysis can be used to find the Purkinje spots, these spots are then removed from the binary image. The highest, lowest, furthest right, furthest left pixels are found in this altered image and used to find the center point and an average radius.

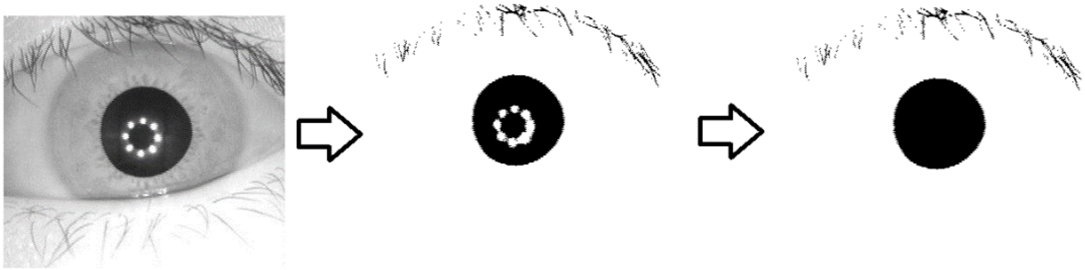

Figure 7 Purkinje spot removal. Base image from [7]

The full grayscale image is again used to find the limbic boundary. First the x and y directional gradients are calculated for the entire image. Next, the sum of each gradient value is found for each radius 1 to the maximum radius of the image. This array will show which radii have the strongest gradient in the image. Theoretically, the two radiuses with the strongest gradients should be the pupil and limbic boundaries. In practice, the strongest four where determined and the largest was used as the limbic boundary.

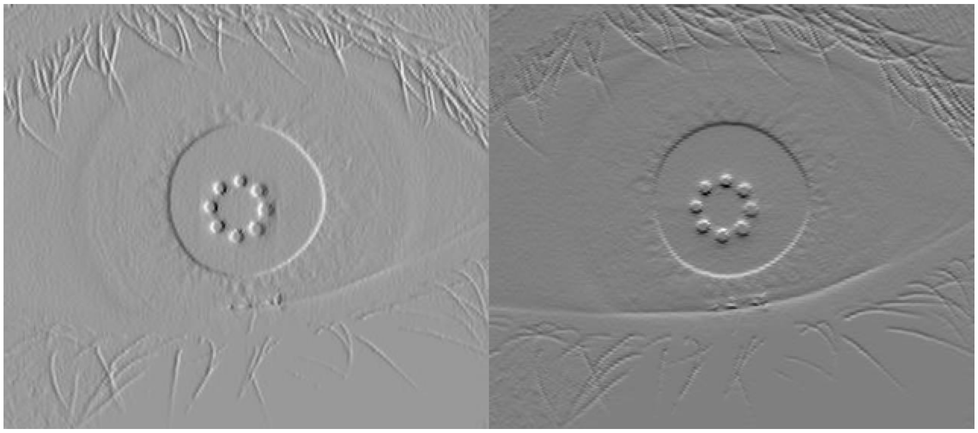

Figure 8 x and y directional gradients for an eye. Base image from [7]

When both boundaries are identified, the image of the iris can be normalized using polar coordinates. The functions defined in [5] were slightly modified and used for the polarization of the image seen in Figure 8. This function initial found the center of the entire image but was altered to use the center of the iris as determined during image localization.

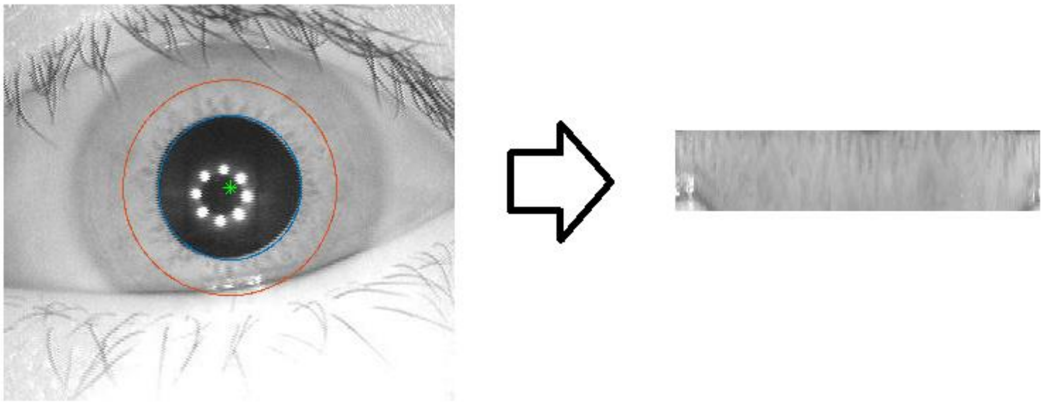

Figure 9 Normalization of section defined between two shown radii

The final step for image processing is feature extraction. A two-dimensional Gabor bank was generated using the functions defined by [6]. The real part of the Gabor filter is shown in Figure 9. After generating a bank of filters this frequency was determined to generate good results and the orientation is set to 90° as [6] listed this orientation as optimal. The included Gabor feature function included in [6] outputs a user specified length feature vector that can be used as the iris code.

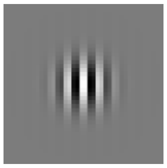

Figure 10 2-D Gabor Filter used

Results of Build

After completing the hardware design for this project, PCBs were manufactured, and parts were ordered. There was difficulty ordering cameras from the manufacturer/distributor within the time and budget constraints of this project. Due to this, the scope of the project had to be altered based on the resources present. Instead of acquiring images with the 1.3 MP NIR camera, a database of irises was used from CASIA [7].

Hardware Manufacture and Testing

Originally the plan for manufacturing was to use solder paste and to reflow the entire board at once. There were several issues with this approach. The first was that the footprints on the PCB for the linear regulator and flash memory had been swapped. All the components besides these two were manufactured in this way. A table top DC power supply was used temporarily in place of the linear regulator to supply 1.8V and 2.8V appropriately. It became immediately apparent that the board had a short from 1.8V to ground and while trying to locate it, several pads were ripped off the board.

A second board was soldered using an iron and tested for shorts intermittently to avoid some of the issues of the first board. Additionally, a breakout board was designed to sit of to the side of the main iris scanner board. This breakout board has the linear regulator on it as well as its required bypass and coupling capacitors. The button was also placed on this board because there was not enough tolerance added on the original board for the pins. Jumper wires were used to connect voltage outputs from the breakout board to header pins or vias on the main board. To connect the flash memory to the incorrect footprint, small gauge wire was soldered to the pads and some vias then soldered to the package. The wires and package were hot glued to the board to keep the connections intact.

The populated PCB is shown below in Figure 10, this board does not include the CPLD or camera interface as these components will not be used without the camera. This board has successfully been verified and some code has been compiled and executed.

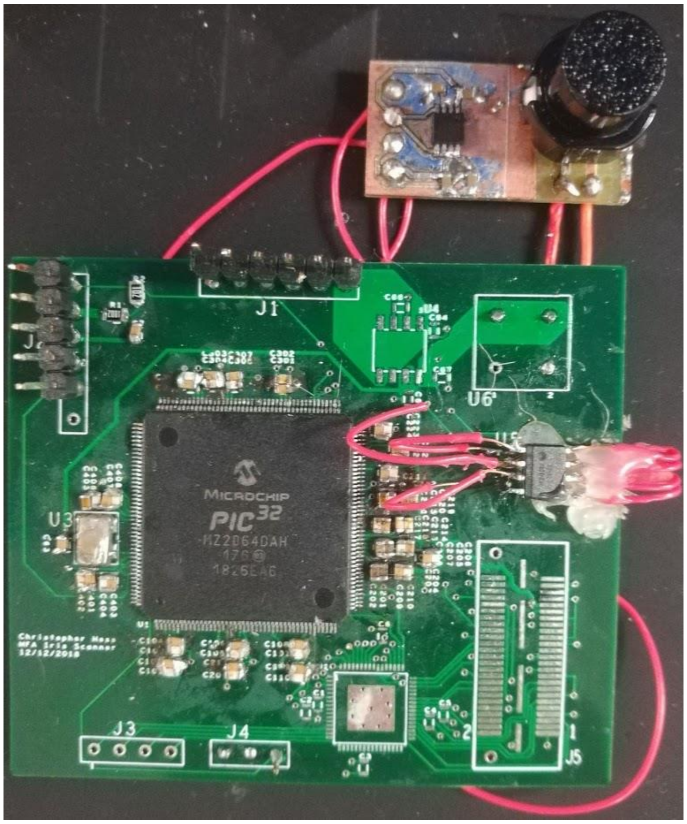

Figure 11 Final PCB

Image Processing Testing

A small subset of eight subject's irises from the CASIA database were used to develop and test the image processing. Each phase of the image processing was tested individually and MATLAB plots were used to visually confirm that each phase was adequately serving its function. Initial results found the hamming distance between the first image of a particular iris against the nine other images of the iris. These results are shown in the histogram below in figure 11. The average hamming distance for this set is slightly higher than expected.

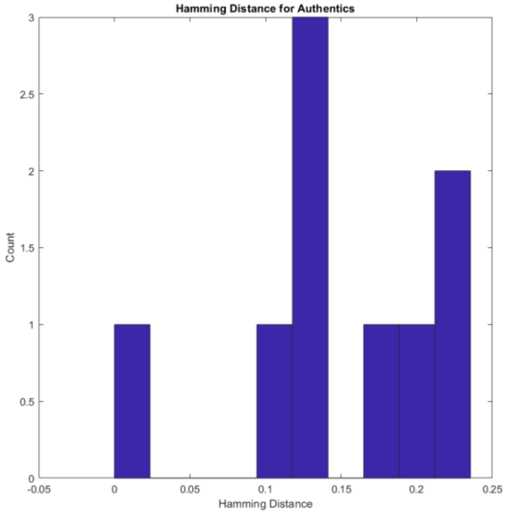

Figure 12 Hamming Distance for Authentics

Additionally, the hamming distance was compared against this same image with nine pictures of other irises. The results of this test are shown below in figure 12. The average hamming distance for this data set is much lower than expected and poses a critical issue for the algorithm. There is not an effective hamming distance threshold that could be set for this data that would have an acceptable accuracy rate. Further analysis was conducted to determine what was causing inconsistence, the biggest issue seems to be locating the limbic boundary. Figure 13 shows three irises with the same limbic boundary locator used.

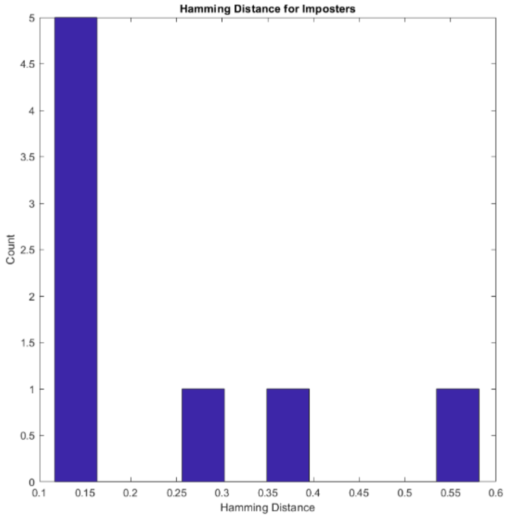

Figure 13 Hamming Distance for Imposters

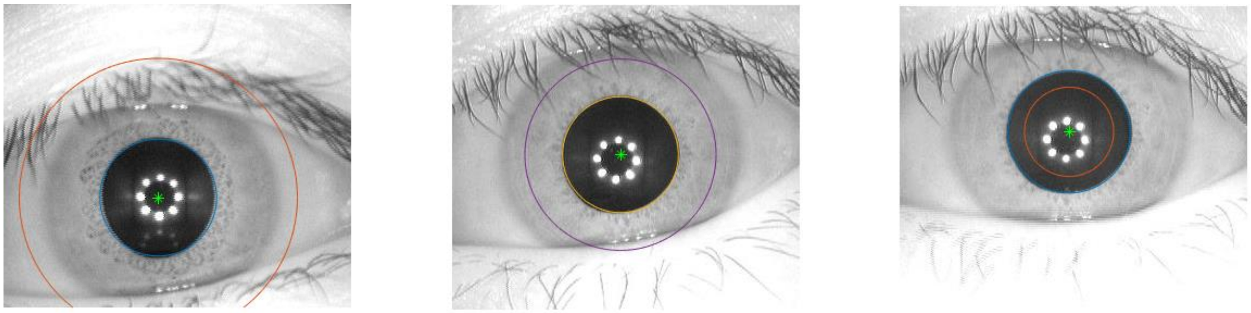

Figure 14 Variance between determined limbic boundaries

Future Work and Conclusions

The iris recognition algorithm needs to be refined before it can be tested on the embedded system. Work for the immediate future of this project should focus on finding a more reliable method for limbic boundary locating. One possible alternative method is to use a Hough circle transform. Analysis will be done to see if this operator works more reliably. Additionally, the low hamming distance for between separate irises suggests that the Gabor filter may need altering. Further analysis will be done to find parameters that are able to extract features more effectively.

Without the selected camera on hand, the functionality of the hardware design for this project has been verified to greatest extent that it can be and the next step is to run the image processing on it. Future work should focus not only on the accuracy of the recognition process, but also the effectiveness of the algorithm on the system. Program execution should be timed and compared to other designs on the market.

References

[1] J. G. Daugman, "High confidence visual recognition of persons by a test of statistical independence," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 15, no. 11, pp. 1148-1161, Nov. 1993.

[2] Y. Si, J. Mei, H. R. Karimi, C. Wang, and H. Gao, “Design and Implementation of a Low-cost Embedded Iris Recognition System on a Dual-core Processor Platform,” IFAC Proceedings Volumes, vol. 45, no. 4, pp. 278–282, 2012.

[3] V. Saravanan and R. Sindhuja, “Iris authentication through Gabor filter using DSP processor,” 2013 Ieee Conference On Information And Communication Technologies, 2013.

[4] PIC32MZ: http://ww1.microchip.com/downloads/en/DeviceDoc/

[5] Polar To/From Rectangular transform of Images. V0.1 16 Dec 2007. Prakash Manandhar pmanandhar@umassd.edu

[6] M. Haghighat, S. Zonouz, M. Abdel-Mottaleb, "CloudID: Trustworthy cloud-based and cross-enterprise biometric identification," Expert Systems with Applications, vol. 42, no. 21, pp. 7905-7916, 2015.

http://dx.doi.org/10.1016/j.eswa.2015.06.025

[7] CASIA Iris Image Database, http://biometrics.idealtest.org/

[8] Kumari, Y. and Sharma, R. (2014). “Iris Recognition System using Gabor Filter & Edge Detection,” International Journal on Recent and Innovation Trends in Computing and Communication, 2(8).