Biometric Data Acquisition using MATLAB GUIs

Robert C. Schultz, Robert W. Ives
U.S. Naval Academy, Electrical Engineering Department
Annapolis, MD 21402-5025 rschultz@usna.edu, ives@usna.edu

Abstract - **The use of iris patterns, fingerprints, facial characteristics and other biometrics for identification is becoming more common every day. This has led to the incorporation of biometric topics in courses at many colleges throughout the world. Developing practical laboratory exercises to augment course lectures is enhanced when the students have the ability to work with actual biometric data. Although the hardware for using biometrics is available at a reasonable cost, most commercial products do not readily output the raw data that is desired for laboratory or research environments. Even if the data can be acquired, a system of storage and retrieval that incorporates unique information related to the data is needed. Finally, the acquisition of biometric data introduces human research and privacy concerns that must be addressed by the institution. Using MATLAB with its Image Acquisition Toolbox, we have developed a system to easily capture biometrics and store them with pertinent information for laboratory exercises, student research and faculty research. We discuss the concerns involving human subjects and privacy, the challenges associated with developing a biometric database, and one solution to acquiring biometric data for use in student laboratories.**

Index Terms - Image Acquisition, Biometrics, GUI, MATLAB

INTRODUCTION

A Biometric is a physiological or behavioral trait which may be measured and subsequently identified in order to confirm an individual's identity [1]. The use of biometrics for human identification starts with user enrollment. Enrollment is typically an overt process where the user voluntarily submits their biometric data, which is then transformed into a biometric template that is stored for future use. Once the user is enrolled, the system can perform verification and identification functions [2].

Verification requires the user to enter some piece of information indicating who they are, such as a username. The system will then capture the biometric and create a new template based on the current information. This template is compared to the template created during enrollment, and if the templates match within a set tolerance, access to the desired resource is granted. On the other hand, identification captures the biometric and compares the template against all users in the system. If the template matches within a set tolerance of any of the enrolled templates, the user is identified and access is granted [2].

Both verification and identification use the biometric template instead of the raw biometric data (such as a captured digital image) to perform these functions. Software Development Kits (SDKs) can be purchased allowing developers to generate the biometric templates, but not necessarily the actual raw biometric data. Even when the ability to capture the raw biometric data exists, it is often encrypted thus limiting its use in the laboratory environment.

At the Naval Academy's Center for Biometric Signal Processing, we have developed a system that interfaces with commercial off-the-shelf hardware and software to capture raw biometric data. Biometric data is sensitive and privacy rights dictate special handling. Since the subjects are human, special precautions must be taken to ensure the process is safe and secure. This paper discusses the human research concerns, the development of the data acquisition system, and the method used to distribute data for student and faculty research.

HUMAN RESEARCH

The collection of biometric data falls into the category of human subject testing. There is an approval process for this type of research at the U.S. Naval Academy that must be followed prior to collecting data, much like there is at many universities or institutions that conduct such research [3]. The primary concern in this case is that because the data that is collected is or could be used for personal identification, it must be protected from those who would use it improperly. Therefore, the privacy of the test subjects must be guaranteed. This section describes the requirements in place to protect the data.

First, the data is segregated: it is maintained only on computers in the biometric signal processing laboratory. The computers that contain the data are not connected to the Internet or the USNA intranet. The custodian of the data is the laboratory director, who maintains a backup copy of the database. The data is not released to individuals outside of those who work in our lab.

Second, the data is anonymous: no name or other identifying information is attached to the biometric data that is stored on the computers. Random characters are used to name the data files of the biometrics collected. A master cross-reference list that associates the biometric data files with an individual is maintained by hard copy in a secure location.

Third, the data is used by students under special circumstances: students who are performing independent research in biometrics are granted access to the data while work1ing in the biometrics lab. They assist in collecting the

data they need for their projects and use the data in their project. At least one of the lab's faculty members is present when collecting any biometric data for the database.

Finally, the subjects who volunteer their biometrics for the database are made aware of the precautions in place and the conditions under which they agree to submit samples. Prior to providing biometrics samples for the database, they read and sign a Consent and Information form, which delineates the purpose, procedures, any risk or discomfort expected, the benefits of the research and describes the conditions to ensure their privacy.

SYSTEM DEVELOPMENT

Acquiring raw data for research can be a long and tedious process. The use of a graphical user interface (GUI) can greatly simplify the process. We have developed a system that captures raw iris images with key parameters that can be used for statistical analysis.

We found the easiest method to implement a capture system was using a device that produced a standard composite video output signal in conjunction with a frame grabber board. The Naval Academy Biometrics Lab contains an LG IrisAccess® 3000 entry control system. The enrollment station contains a high quality near infrared camera that automatically maintains focus on the iris provided the subject is within a certain range. The camera used to acquire the images also uses a composite video output. Using a BNC T-Connector we route the signal output of the camera to the enrollment station and also to a frame grabber board on a separate computer that is used for data collection (Figure 1). The data collection computer has a Matrox Meteor II frame grabber configured with MATLAB 7.0 and the Image Acquisition Toolbox.

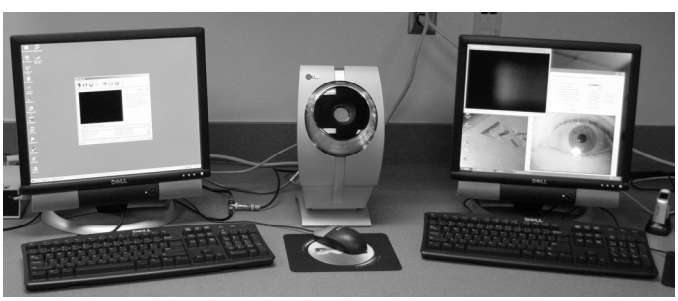

FIGURE 1
IRIS CAPTURE HARDWARE

I. GUI Layout

At the USNA, MATLAB was our language of choice due to its ease of use resulting in rapid project turnaround times. MATLAB provides a GUI Layout Editor that is well integrated with their editor and debugger. Each GUI has a MATLAB figure and an m-file associated with its file name. The MATLAB figure provides the layout of the GUI and the m-file provides the functionality. The Layout Editor is started using the *guide* command. This provides a dialog offering a number of designs to choose from. Selecting the default template is sufficient and results in Figure 2.

The left panel of the layout editor provides all the components found in a typical Windows application. Determining which of these components to use varies with the project. At least one set of axes must be added to display data in graphical format. The data we collect requires a combination of text boxes, check boxes, axes, and buttons. The data and the types of objects used to collect the data are listed in Table 1. The type of objects selected is based on the information collected. Subject Number, Iris Age, and Image Number all require user input and the range of values is unknown until the time of data collection, thus we chose an Edit Box to collect that information. We chose to limit the options available for Iris Color so a Popup Menu was selected to display available options. Gender and Eye have two mutually exclusive options thus a Grouped Radio Button was our choice of data collection. The remaining data was not mutually exclusive so our choice was a series of Check Boxes grouped together.

TABLE 1
GUI DATA TYPES AND OBJECTS

Description	Data Type	GUI Object
Subject Number	Numerical String	Edit Box
Eye	Boolean	Grouped Radio Buttons
Gender	Boolean	Grouped Radio Buttons
Iris Age	Integer	Edit Box
Iris Color	Multiple Choice String	Popup Menu
Image Number	Integer	Edit Box
Amplifying Data	Boolean	Grouped Check Box
Image Source	Multiple Choice String	List Box
Image Acquired	Bitmap	Axes

The remaining objects for the GUI are used for program control. It is possible to have multiple data sources on any given machine. Our data acquisition computer has multiple web cameras, a fingerprint reader, and a frame grabber. The Image Source List Box allows us to choose the source of data. We populate the control object with all available devices when the program is started allowing the user to select the desired data source. The push buttons allow the user to preview the data, capture a single image, capture a series of images and store the images.

Each object that is placed on the layout has numerous properties associated with it. A property inspector allows the programmer to adjust properties as desired. One property that is important to set and know is the Tag property. The Tag is the noun name given to the object so that it can be accessed inside the m-file. To aid programming, we set each object's Tag name so that it can be easily identified inside the corresponding m-file. For example, the text box that contains the iris age has the Tag name *IrisAge.*

II. Controls

GUIs are event-driven programs. When an event such as a button click occurs, the program executes a callback function.

The callback function is passed the handle to the object, information regarding the event and a structure containing the handles for all the objects in the GUI. Initially, our system was designed to capture a single image at a time. This was implemented using a Preview Button, a Capture Image Button, and a Save Image button. The quality of the images acquired required careful timing to prevent blurriness caused by delay times in the auto focus mechanism. This frequently resulted in numerous attempts to acquire a single high quality image.

To ease the capturing process, we modified the software in two ways. First we added a preview function allowing the operator to verify the subject is properly situated prior to activating the capture routine. Next we added a function that captures ten frames of video over a one second time interval. We found this interval provided the desired number of images at sufficient quality. Prior to storing the data, both the single image and multiple image capture routines display the results on the GUIs axes. This allows the operator to evaluate image quality prior to storing the data. If the operator observes poor quality images, they simply re-click the capture button and the new results are displayed.

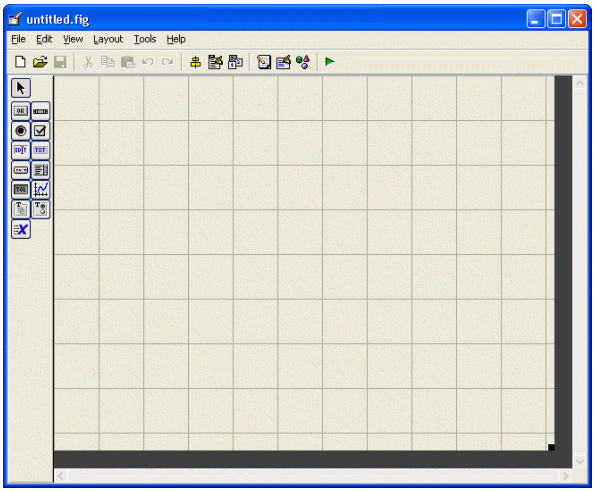

FIGURE 2
DEFAULT GUI LAYOUT

III. Image Acquisition

The images were collected using the Image Acquisition Toolbox in conjunction with the Matrox Meteor II frame grabber. The MATLAB Image Acquisition Toolbox provides an interface for image acquisition devices made by Coreco Imaging, Inc., Data Translation, Inc., and Matrox Electronic Systems, Ltd. Additionally, the toolbox supports devices that provide a Windows Driver Model (WDM) or Video for Windows (VFW) driver. Finally, the toolbox supports devices that comply with the IIDC 1394-based Digital Camera Specification (DCAM) [4]. We chose a frame grabber based on the composite video output of the iris camera equipment connected to our enrollment station.

The Image Acquisition Toolbox provides a number of ways of collecting image data via the frame grabber. We use two different methods in our application. The first uses the *getsnapshot* function. When executed, it simply returns a matrix containing a single frame of image information present at the time of execution. The second method uses triggering.

As previously mentioned, we found it necessary to capture short video clips to ensure quality images were obtained. We implemented this function using the triggering functionality of the input video object. We set the video object's *FramesPerTrigger* property to the number of frames to be captured. Next we set the *FrameGrabInterval* to three. A *FrameGrabInterval* of three tells MATLAB that every third frame should be captured. The operator initiates video capture by clicking on the Capture Video button. This executes the Image Acquisition Toolbox *start* command and begins storing the data in a private buffer related to the video object. During the capture process, the user can not interact with the program. Once the capture process is complete, program control is restored and the data is transferred to the MATLAB workspace [4].

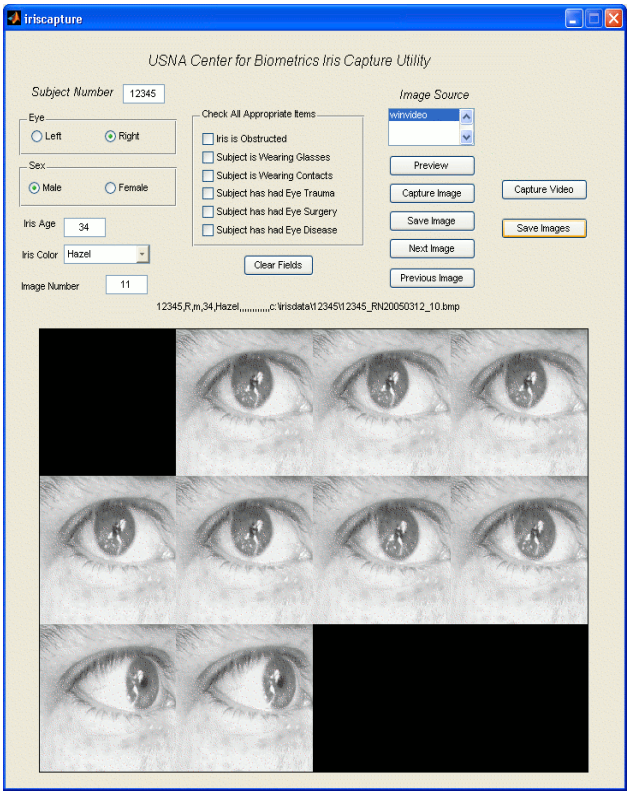

FIGURE 3
IRIS CAPTURE UTILITY LAYOUT

The data is returned to the MATLAB workspace as a four dimensional array. The first two dimensions correspond to the rows and columns of each individual frame. The third dimension corresponds to the Red, Green, and Blue color components. The final dimension corresponds to the frame number (relative time). The software then uses the *imaqmontage* command to display the images for user quality

verification as shown in Figure 3.

Figure 3 also illustrates the usefulness of obtaining multiple images. These images were collected for demonstration purposes only using a low cost web camera that supports Video for Windows. Note that the subject moved his eye during the capture process. This sequential data can be used to analyze liveness testing algorithms in addition to position tracking algorithms.

The final stage of the image acquisition process is storing the data. After the operator verifies that the images are of sufficient quality, he or she will click the Save Images button. This directs the program to automatically generate the filenames based on the subject's random code and store the information as described the following section.

IV. Storage and Retrieval

Raw biometric data obtained via a video capture device requires large quantities of storage space. A standard 640x480 image acquired through the Matrox frame grabber requires 302KB of storage space. In this study, each subject has multiple images taken of each iris with and without glasses and contacts. Additional images are taken at a later date to analyze the effects of changes over time.

Advances in storage technology have resulted in relatively inexpensive large capacity hard disk drives. The data acquisition computer is equipped with a 120GB hard disk drive and a DVD recorder. Each DVD can store approximately thirteen thousand images. This equates to a maximum of just over six hundred subjects per DVD. Typically the images are stored in a secure folder on the hard disk. Following a data acquisition session with new subjects, the data is backed up to a DVD and stored in a secure location.

Generating a unique filename that allows researchers to efficiently use the data was a significant concern for this project. Verifying the performance of biometric algorithms requires knowledge of which iris images come from the same eye. With the large number of images planned for collection, filenames were employed that allowed rapid confirmation that a group of irises were from the same subject. Additionally, it was important that some directory structure be maintained to avoid having thousands of files in a single directory. The following filename standard was used. The first five characters are the random characters that identify the subject. This is followed by either the letter "L" or "R" identifying if it is the right or left eye. The next character is a "N", "G", or "C" identifying whether the image is without glasses or contacts, with glasses, or with contacts respectively. Finally the filename concludes with the date the image is taken and the image number indicating the relative order in which the picture was acquired that date. The images are than stored in a folder corresponding to the random characters that identify the subject.

Amplifying information is stored in a comma delimited text file. An entry is made for each image collected. A given entry will contain all the fields specified in the GUI, the filename, and the location of the file. The use of a comma delimited text file allows easy incorporation into a database or spreadsheet for later use. Additionally, it can be searched to obtain a list of all iris images that have a certain set of characteristics. For example, a group of images where the subject had eye surgery after the initial capture could be selected and used to determine how a particular algorithm or set of algorithms perform given that a subject had eye surgery after enrollment.

STUDENT APPLICATIONS

Since the initial establishment of the database, the data has been put to use in a number of student projects and learning activities. One student used the data to assist in developing an algorithm for locating the iris of non-orthogonal iris images. Another used the system to aid in developing an algorithm to enhance the search speed for iris matching. His results were presented at the National Conference for Undergraduate Research [8].

The GUI was also used to demonstrate a system that uses real time video capture to seniors whose final projects required similar techniques. This aided their understanding of the requirements for real time data capture and manipulation.

Finally, we intend to integrate a similar GUI in future biometrics courses to provide an active-learning environment. One laboratory exercise will allow the students to capture their own fingerprint biometric data and have them to manipulate and identify unique features contained in the data.

FUTURE ADDITIONS

Currently the software requires the operator to verify the quality of the images. The Image Acquisition Toolbox provides the resources to analyze each frame as it is acquired. Future versions will provide the option to collect and analyze the data frame by frame using an algorithm of the operator's choice.

In addition to iris images, collection of fingerprint and face biometric data is desirable. We are currently in the process of developing a system to simultaneously collect face, finger and iris biometrics. It is expected that this will provide many opportunities for continued research in biometrics by students as well as faculty [5]-[8].

CONCLUSION

It is clear that laboratory projects can assist a student's understanding of biometric systems. In order to perform these projects, a source of biometric data must be available. It is also useful if the data can be accessed using a system that does not cause the students or researchers to be bogged down in programming nuances. MATLAB, its Image Acquisition Toolbox, and the GUI Layout Editor provide this environment. The sensitive nature biometric data requires a conscious effort to maintain privacy of the subjects. We have presented a system that acquires unique iris biometric data for use in research and learning activities.

REFERENCES

[1] Ashbourn, J, "*Practical Biometrics From Aspiration to Implementation*", Springer, 2004

[2] BioAPI Consortium, "*BioAPI Specification Version 1.1*", http://www.bioapi.org/, 16 Mar 2001.

[3] "USNA Institutional Research", http://www.usna.edu/IR/htmls/proposal_instructs.html, 19 March 2005

[4] "MATLAB Image Acquisition Toolbox Help", http://www.mathworks.com/access/helpdesk/help/toolbox/imaq/imaq.shtml, 19 March 2005

[5] Ives, R, W, Du, Y, Etter, D, M, Welch, T, B, "A Multidisciplinary Approach to Biometrics," *IEEE Transactions on Education*, Dec 2004.

[6] Ives, R, W, Bonney, B, Etter, D, M, "Effects of Image Compression on Iris Recognition," to be presented at the 2005 IEEE Instrumentation and Measurement Technology Conference, Ottawa, Canada, May 2005.

[7] Du, Y, Ives, R, W, Bonney, B, Etter, D, M, "Analysis of Partial Iris Recognition," *2005 Biometric Technology for Human Identification II Proc. SPIE Vol. 5779*, p. 31-40

[8] Horwath, J, Ives, R, W, Etter, D, M, "Smart Search Iris Recognition," *19th National Conference on Undergraduate Research*, Apr. 2005.