Real-time Web-based Hyperspectral Data Viewing and Modeling

GRADUATE PROJECT REPORT

Submitted to the Faculty of
the School of Engineering and Computing Sciences
Texas A&M University-Corpus Christi
Corpus Christi, TX

in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Computer Science

by

Linh Manh Pham
Fall 2011

Committee Members

Dr. Mario Garcia
Committee Chairperson

Dr. Ruby Mehrubeoglu
Committee Co-chairperson

Dr. David Thomas
Committee Member
ABSTRACT

There is much interest in hyperspectral data in multiple disciplines. The analyzed samples can vary from plants and agricultural products, to algae, coral and metals. A Hyperspectral Imaging System (HIS) has been set up and used to capture hyperspectral images from various samples in the 400-1000 nm spectral region. The system consists of an imaging spectrometer attached to a CCD camera with a fiber optic light source as the illuminator. The LabVIEW-based software is used to interface with the system, to collect and display the data. The significance of this system is its capability to capture 3D spectral and spatial data that can be analyzed to extract both conceptual and spectral information about the underlying samples. The challenge with this data remains with the large data size and time-intensive data processing of hyperspectral image cubes, which can be in the order of gigabytes if collected at full resolution.

This report describes interfacing the HIS with the Internet through NI LabVIEW and related programs so that the system can be controlled remotely and the data can be viewed real-time and on-line by individuals who are part of a research project but who are not in the lab. The information provided by the system can then be used to calculate the spatial, spectral and optical properties of the samples by applying mathematical methods. For these goals, a model which supports remote acquisition and analysis of hyperspectral image has been proposed with the Web-based LabVIEW interface using the NI Internet Toolkit and NI Web UI Builder. By this interface, authenticated users can access and control the HIS system remotely. Experimental results are stored in a FTP server which serves the purposes of querying analyzed data and viewing the images from the Internet to avoid downloading limitations of large datasets.
# TABLE OF CONTENTS

Abstract ......................................................................................................................... ii

Table of Contents .......................................................................................................... iii

List of Figures ................................................................................................................ v

List of Tables ................................................................................................................ vii

1. Introduction and Background .................................................................................. 1
   1.1 LabVIEW-Based Web Interface ........................................................................ 2
   1.2 Related Work ...................................................................................................... 3

2. Hyperspectral Imaging System Models ..................................................................... 5
   2.1 Local Control HIS Functional Model ................................................................ 5
   2.2 Remote Control HIS Functional Model .............................................................. 6

3. HIS Hardware System .............................................................................................. 8

4. Web-based LabVIEW Application Interfacing the Lab-based HIS ......................... 11
   4.1 Graphical User Interface .................................................................................... 12
   4.2 The Graphical Program ..................................................................................... 17
   4.3 The Website ........................................................................................................ 20

5. Data in the Remote Control HIS ............................................................................ 26
   5.1 Data Acquisition ............................................................................................ 26
   5.2 Data Storage .................................................................................................... 28
   5.3 Data Viewing with the Web-based HIS ............................................................. 32
   5.4 Data Sharing with the Web-based HIS .............................................................. 37

6. Testing and Evaluation ............................................................................................ 40

7. Future Work and Conclusions ................................................................................. 43
7.1 Future Work ........................................................................................................43

7.2 Conclusions ........................................................................................................44

Bibliography and References ..................................................................................45

Appendix A – Graphical Diagram .........................................................................48
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Local Control HIS Functional Model</td>
<td>6</td>
</tr>
<tr>
<td>2.2</td>
<td>Remote Control HIS Functional Model</td>
<td>7</td>
</tr>
<tr>
<td>3.1</td>
<td>Network Topology of the Real-time Remote HIS</td>
<td>9</td>
</tr>
<tr>
<td>3.2</td>
<td>The Components of the Local Control HIS</td>
<td>10</td>
</tr>
<tr>
<td>4.1.1</td>
<td>Front Panel of the Start-up Program</td>
<td>13</td>
</tr>
<tr>
<td>4.1.2</td>
<td>Front Panel of Application 1</td>
<td>14</td>
</tr>
<tr>
<td>4.1.3</td>
<td>Front Panel of Application 2</td>
<td>15</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Graphical Diagram of the Start-up Program</td>
<td>18</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Graphical Diagram of Application 1</td>
<td>19</td>
</tr>
<tr>
<td>4.2.3</td>
<td>Graphical Diagram of Application 2</td>
<td>20</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Web-based User Interface of the Homepage</td>
<td>22</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Web-based User Interface of the Start-up Page</td>
<td>22</td>
</tr>
<tr>
<td>4.3.3</td>
<td>Web-based User Interface of Application 1</td>
<td>23</td>
</tr>
<tr>
<td>4.3.4</td>
<td>Web-based User Interface of Application 2</td>
<td>23</td>
</tr>
<tr>
<td>5.1.1</td>
<td>Setup the Camera Graphical Diagram</td>
<td>28</td>
</tr>
<tr>
<td>5.1.2</td>
<td>Record Images Graphical Diagram</td>
<td>28</td>
</tr>
<tr>
<td>5.3.1</td>
<td>Left Screen of Application 2</td>
<td>32</td>
</tr>
<tr>
<td>5.3.2</td>
<td>Right Screen of Application 2</td>
<td>33</td>
</tr>
<tr>
<td>5.3.3</td>
<td>Viewing Spreadsheet Online</td>
<td>34</td>
</tr>
<tr>
<td>5.3.4</td>
<td>Sending Hyperspectral Data Using FTP</td>
<td>35</td>
</tr>
<tr>
<td>5.3.5</td>
<td>Real-time Camera Screen in Application 1</td>
<td>36</td>
</tr>
</tbody>
</table>
Figure 5.3.6. Graphical Diagram of Outputting Real-time Spectral Images .................36
Figure 5.4.1. Web-based FTP Client Authentication Page....................................38
Figure 5.4.2. Web-based FTP Client Page after Authentication............................39
Figure 6.1. FTP Server Command Line.................................................................41
Figure 6.2. Getting a File Using FTP Command.....................................................41
# LIST OF TABLES

Table 3.1. Network Configuration of the Real-time Remote HIS ........................................9
Table 4.1.1. Configurative Parameters ..............................................................................17
Table 5.1.1. Recorder Mode vs. FIFO Buffer Mode ..............................................................26
Table 6.1. FTP Transfer Rate ...............................................................................................42
1. INTRODUCTION AND BACKGROUND

The development of tools in modern science and technology and the accessibility of these tools are very important for educational development. The learning process can be improved by various methods. New concepts in education are provided by many technological researchers and developers [Baser 2006], [Bauer 2007] and [Ertugrul 2000]. The explosion of the Internet facilitates has resulted in new approaches and new methods to design and develop virtual and real-time laboratory applications that benefit educational purposes. Laboratory work is one of the important elements for researchers and students. New students or young researchers can practice what they learned in their classes [Salzmann 2000] through laboratory experience. The growth of the number of students, however, limits the number of students who can use the laboratory facilities. Moreover, implementation of a laboratory which can meet the required standards has generally a very high price [Baye 2001].

The virtual lab (VL), which can be controlled in real-time and remotely, is another way to resolve time, location and cost issues related with physical labs. This trend is taking over the traditional laboratory. In virtual laboratories, students have access to the experiments remotely anytime and anywhere with any devices they have [Baser 2004]. In addition, students can set up parameters in the virtual devices as well as repeat the experiments from a remote location. This kind of laboratory reduces the requirements for hardware and software. The remote users do not have to buy underlying hardware to setup the laboratory as well as dedicated software to control devices in the labs.

Hyperspectral imaging (HSI) refers to the collection and processing of data information from across the electromagnetic spectrum [HIS 2011]. Human eyes can only
see visible light in three basic bands (red, green, and blue). Spectral imaging divides the electromagnetic spectrum into many consecutive bands which can go beyond the visible range. Many hyperspectral sensors and processing systems are created to provide the capability which can look at objects in a portion of the spectrum. Certain substances leave uniquely spectral signatures across the electromagnetic spectrum. Those signatures enable identification of the materials which form a scanned object. For example, miners may find a new gold mine based on a spectral signature for gold [Werff 2006].

Laboratory-based hyperspectral imaging has found applications in many disciplines because of the spectral, spatial, and temporal information it can provide under controlled conditions. Analyzed samples can vary from biological cells to metallic surfaces. The hyperspectral image cubes are acquired manually where the user adjusts acquisition parameters in the lab. The hyperspectral imaging experiments are usually performed in a dark-room environment, which may not be comfortable for many users at extended periods of time. Moreover, distant users who may not have easy access to the system are not in a position to provide immediate feedback on the acquired data. The scarce hardware research resources along with dedicated as well as complicated software that is available in the HSI lab put much burden on a new user who wants to join in a highly time-constraint project. The disadvantages of a physical HSI laboratory lead to the demand for a virtual Hyperspectral Imaging System (HIS) where the users can control the HIS remotely in real-time manner.

1.1 LabVIEW-Based Web Interface

LabVIEW is one of the high-level graphic programming languages, which is often chosen to build the VL laboratory. LabVIEW software provides tools for building a
virtual laboratory with capabilities that range from testing, instrument control, and measurement, to data processing and data analysis [Higa 2002]. The virtual instrument can be designed and controlled via GPIB, RS232 or VXI interfaces. The Web-based applications for users to access a virtual laboratory by the Internet can be built using the NI Internet Toolkit and NI Web UI Builder. The LabVIEW program provides both logical and physical visualization of devices as well as their simulations [LVM 2010]. LabVIEW also has a built-in Web server that not only supports regular world-wide Web languages such as HTML, and XML but also supports dedicated LabVIEW modules. Those modules provide the ability to simulate objects in real life, thus digital devices or even mechanical machines are imitated efficiently. These objects are called Virtual Instruments (VIs). The capability of drawing real-time waveform charts is another advantage of LabVIEW, especially for plotting hyperspectral graphs. The NI LabVIEW Internet Toolkit provides a variety of popular network application-layer protocols such as HTTP, FTP, or SMTP. Moreover, popular applications providing remote control services like VPN, VNC or Logmein require manual set-up software in the client side (client agent). With LabVIEW, run-time modules integrated into the client browser as a plug-in is installed in the client machine automatically while a HTTP header is being called [LRM 2010].

1.2 Related Work

Many Web-based virtual laboratory examples are proposed in the literature and implemented in practice. Some of them are discussed in this proposal. In [Cheng 2002], a power electronics experiment is simulated in a remotely controlled laboratory. In this lab, a power system is controlled and monitored by a Web-based tool. In [Ko 2001], the
National University of Singapore (NUS) introduces a new Web-based virtual laboratory to simulate frequency modulation experiments in an undergraduate course. In [Xiaoyan 2005], an electrical and electronics virtual laboratory that describes and shows up aspects of instruments and circuits is also used for teaching. LabVIEW was selected for the simulation. In [Dormido 2007], a Web-based distance learning platform is performed. This platform benefits the course “Electrical Measurements” at Faculty of Electrical Engineering in Zagreb, Croatia.

Most of the remote laboratories involve pre-designed experiments with cook-book type answers to teach and learn predetermined concepts. In this paper, we describe access to a laboratory-based hyperspectral imaging system (HIS) set up for research purposes. Typically, a lab-based HIS operates under dark-room conditions to avoid ambient light contaminating the controlled light source. Hyperspectral images are complex and require interdisciplinary expertise for analysis and interpretation. The motivation for a remotely accessed and controlled HIS arose from the need for the collaborators of the HOPI Lab team to access the data real-time and easily to select optimal parameters for the media to be tested. Hyperspectral image cubes can be over 3 GB in size if taken at full resolution, and, hence, limits the feasibility of fast data transfer for analysis. With the remote access and control, data can be captured and viewed on the host computer without transferring the data on the Internet.
2. HYPERSPECTRAL IMAGING SYSTEM MODELS

This project describes interfacing the HIS with the Internet through NI LabVIEW 2010 and related programs so that the system can be controlled remotely and the data can be viewed real-time and on-line by individuals who are part of a research project but who are unable to access the lab physically. The information provided by the system can then be used to calculate the spatial, spectral and optical properties of the samples by applying mathematical methods. Moreover, concerns of security are taken into account in order to guarantee a clear and safe shared-data environment, especially on the Internet. To complete these goals, the project is divided into 2 parts that has been done in separate phases: A Web-based HIS functional model and a LabVIEW application interfacing Lab-based HIS. This section focuses on functional models of hyperspectral imaging systems.

2.1 Local Control HIS Functional Model

The applications of hyperspectral data in multiple disciplines mentioned in Chapter 1 raises demands for the design of an interface for the hyperspectral imaging system to collect various hyperspectral image samples from different sources and then analyze, store as well as share the data. So far many endeavors have been pursued to make this desire come true in respect of both local and remote control systems. Efforts, that can be found in [John 2003] and [Jeremy 1999], in local-control HIS include information in terms of both system modeling and application development. One of the hyperspectral imaging systems implemented at TAMUCC comes from Headwall Photonics, a company operating in the optical industry. The diagram of the local control HIS is shown in Figure 2.1. Functional model of the local control HIS, which shows what
the system can do, is depicted in this figure. Hyperspectral images can be obtained in the Data Acquisition module and sent directly to the Data Storage module for storage (I, red). The obtained hyperspectral images can be sent to the Data Analysis module for image processing. The result can be saved through the Data Storage module (II, black). The stored data can be retrieved from the Data Storage module, sent to the Data Analysis module for image processing, and then sent back the Data Storage module (III, blue). Data can be viewed from Data Acquisition, Storage or Analysis modules directly (IV, green).

![Diagram of Local Control HIS Functional Model]

**Figure 2.1: Local Control HIS Functional Model**

### 2.2 Remote Control HIS Functional Model

In comparison with the local-control HIS, research in the remote-control HIS has not been as widely published. Some researchers have used local-control HIS to solve problems under more controlled conditions. A couple of models can be found in [Vladimir 2006], and [Krehbiel 2003]. In spite of the limited research in remotely controlled hyperspectral imaging systems, demands for a remote-control HIS is high because of the reasons discussed in Chapter 1. With respect to research collaboration, the local lab-controlled system cannot satisfy increasing demands of sharing hyperspectral
data due to limitations on system usage in the physical lab, as well as the sheer size of the hyperspectral images. As a solution, a remotely controlled HIS model can overcome the disadvantages of a locally controlled HIS. This project proposes a preliminary seven-module remote-control HIS model shown in Figure 2.2. With this model, the user can complete the HSI process from the beginning to the end automatically. The particular modules of the process from user management, system security to data acquisition, data storage, data analysis, data viewing and data sharing are performed remotely and programmatically in a real-time manner. The data from the Data Acquisition, Analysis, Storage modules can be sent to the Data Sharing module for online sharing (IV, purple) or to the Data Viewing module for online viewing (IV, green). The user management module keeps track of multiple users who access and control the system. The security module affects every other module to ensure secure access to the system.

**Figure 2.2: Remote Control HIS Functional Model**
3. HIS HARDWARE SYSTEM

At TAMUCC, a Headwall Photonics hyperspectral imaging system is installed in Hyperspectral Optical Property Instrumentation (HOPI) laboratory as part of NSF-supported project (NSF grant #0960000). The HOPI project aims to develop and test the research instruments which are used to measure the hyperspectral properties of materials. The network topology of the real-time remote HIS is shown in Figure 3.1. LabVIEW and Apache Web service are installed in a computer playing a role as a central server which provides Web-based interface for the HIS system. The TAMUCC Computer Services has assigned an internal static IP address to this computer: 172.x.y.z. This IP address is reserved for the central server; thus whenever the central server joins TAMUCC’s campus network, the IP address is mapped to MAC address of the central server. The internal IP address is translated to a static global IP address by Network Address Translate protocol (NAT). This static global IP address 64.x.y.z is also assigned by TAMUCC Computer Services. Finally, the global IP address is mapped to domain name hopilab.tamucc.edu where a Website of the remote control HIS is published. The central server is connected to a cluster of FileZilla FTP servers (version 0.9.4) located on the same network. This cluster has a common IP address 10.x.y.z. The hyperspectral images and analyzed data are sent to the FTP cluster for storage. FTP service runs on these storage devices to serve transmitting images and data between the local system and the remote user. The size of hyperspectral image is often huge and can be in the order of several gigabytes. The local control HIS is directly controlled by the central server through serial cables. The details of network configuration are shown in Table 3.1.
Figure 3.1: Network Topology of the Real-time Remote HIS

Table 3.1: Network Configuration of the Real-time Remote HIS

<table>
<thead>
<tr>
<th>Device Name</th>
<th>IP Address</th>
<th>Subnet Mask</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Central Server</td>
<td>172.x1.y1.z1</td>
<td>x.y.z.u</td>
<td>Web Service/MySQL</td>
</tr>
<tr>
<td>Storage Server</td>
<td>10.x2.y2.z2</td>
<td>x.y.z.u</td>
<td>FTP Service</td>
</tr>
<tr>
<td>Workstation</td>
<td>10.x3.y3.z3</td>
<td>x.y.z.u</td>
<td>Testing Computer</td>
</tr>
</tbody>
</table>

In Figure 3.2, the local control HIS components located in the HOPI laboratory are shown. In this hardware structure, the MICOS stage controller is connected to the central server through the RS232 cable. An adapter is also used to convert RS232 interface to the COM interface supported by the central server. LabVIEW is adopted to implement the local instrument control. On the basis of the RS232 address assigned to the instrument, the commands are transferred through the cable to the corresponding RS232 box which implements the commands and executes the actions. These actions are the mechanical movement of the stage controller. The central server implementing LabVIEW and Ethernet interfaces communicates in a full duplex form with the HIS. The full duplex form allows communication in both directions, which brings forward faster transmission.
The Cooke PCO.1600 camera is connected directly to the central server through a regular FireWire cable (IEEE 1394). Both the imaging data and control signal are sent back and forth in the same cable. It is a smart CCD Camera equipped with functional modules such as camera RAM (camRAM), camera processor (COC processor), interfaces (FireWire, Camera Link or Gigabit Ethernet), similar to a regular PC in miniature.

Figure 3.2: The Components of the Local Control HIS [Mehrubeoglu 2011]
4. WEB-BASED LABVIEW APPLICATION INTERFACING

THE LAB-BASED HIS

A Web-based application developed totally on LabVIEW platform has been implemented after the theoretical aspect of the work on the remote-control HIS model which had been completed by the end of Phase 1. The application provides a Web-form interface running on LabVIEW Web Server 2010 and Apache Web Server, and supports remote acquisition and analysis of hyperspectral images. By this interface using the NI Internet Toolkit, authenticated users can access and control the HIS system remotely using only a regular Internet-connected computer. The experimental results are stored in a FTP server which serves the purposes of querying analyzed data and viewing the images from the Internet. The LabVIEW FTP client has been developed to serve goal of the project. The development of the HIS Web-based application through this project hides the underlying complicated hardware and software as well as eliminates the need to install and run the application on the client’s own computer. This partly fulfills requirements of Software as a Service (SaaS) in terms of Cloud Computing [Armbrust 2009].

An application that comes with the Headwall Photonics HIS developed on LabVIEW 8.2.1 by the same company [HSK 2010] is used for image acquisition in the lab. This software allows users to collect and view hyperspectral data of limited size in case researchers are actually present in the laboratory and in front of the HIS. In the next subsections, the details of the LabVIEW applications are described from user interface design to structure of source code and the Website.
4.1 Graphical User Interface

The HIS Web-based application has a total of four user interface pages listed as a user authentication page, a start-up page and two separate application pages. In this section, the front panels of the start-up page and real-time applications page developed in LabVIEW are presented. The user authentication page is developed by PHP and runs on Apache Web server. The front panels are called the user interface. These front panels provide a friendly user interface and the required control functionality. The front panels are divided into two sections: control buttons and output screens. The control buttons are used to configure the set-up parameters of the camera, the horizontal stage controller necessary for the 2D image capture, or the entire system. The output screens show the results corresponding to input parameters of system. The results can be shown in a waveform graph or in a real-time camera screen. The front panels are designed in a simple and intelligent structure which follows top-down design with the control buttons section above the output screen section. The purpose of Application 1 is to acquire the hyperspectral images using line scanning camera technology. Application 2 is a tool used to view the stored hyperspectral data.

In Figure 4.1.1, the front panel of the start-up page is shown. In the start-up page, the preliminary configuration interface of two applications is presented. With Live Camera and Live Acquisition, the user selects the communication port which is the interface between a central server and the stage controller. To set up the system, a sample needs to be placed on motion-controlled stage. The stage (MICOS) movement is controlled by the
central server. The purpose of the stage controller is to move the sample in the horizontal direction under the camera and spectrograph. This motion is required for push-broom hyperspectral imaging systems to build the spatial dimension of the hyperspectral images.

With *Viewing Hyperspectral Image and Spectral Plot*, the user at minimum must select the hyperspectral file to be opened. These files have been stored in a shared folder where authenticated users can save their BIL (band interleaved by line) and HDR (header) files. A list of files in the shared folder is also provided to remind the users of the hyperspectral file names. Because the two applications cannot run simultaneously, after the users are finished with one of the applications, they must restart and also refresh the system. Then they can initialize the other application. A **Refresh** button is provided to restart the

---

**Figure 4.1.1: Front Panel of the Start-up Program**
applications as well as clearing the system. The front panels for Application 1 and Application 2 shown in Figures 4.1.2 and 4.1.3 respectively are described next.

![Front Panel of Application 1](image)

**Figure 4.1.2: Front Panel of Application 1**

In *Live Camera and Acquisition* page (Figure 4.1.2), the parameters of camera as *Number of Frames to Average, Spectral Binning, Integration time ms, Collect reference* or *dark reference*, are set in real-time by using the appropriate control buttons; therefore, the users can change the quality of output hyperspectral image in real-time through the change of these input parameters, view the results and readjust the parameters as necessary. A control panel (top half of Figure 4.1.2) including control buttons of the stage is also presented in Application 1’s front panel. The users can specify *Start position, Scan length and Step size* of the stage controller. The absolute position of the stage controller can be adjusted by *Target* parameter. The options for saving BIL and HDR files, high-
reflectance reference image as well as dark room images with all light sources off are designed logically and intuitively. The dark room images are used during image normalization to subtract from the sample images as DC offset noise. The reference image is used to capture the response of the silicon imager in the CCD camera for sample hyper spectral image normalization also. A screen shows the real-time spectral response of the line scan which is being captured and recorded by the camera. When the user scans the sample, the current position of the stage is updated in the position box. Also the percentage of scanning is reported real-time. When the user activates the real-time screen, a pointer will be shown up in this screen. This pointer points to a particular pixel in the screen. The user can select another pixel using four buttons **Up**, **Down**, **Left** and **Right**. The information of the selected pixel is presented in the main screen as well as in two secondary screens which are on the left and under the main screen.

![Figure 4.1.3: Front Panel of Application 2](image-url)
In *Viewing Hyperspectral Image and Data Plots* (Figure 4.1.3), the users can choose viewing mode among *Scan through bands, Grey Image* or *RGB image*. The speed of reading the images is also included by adjustment of *Scan band step* and *Scan Delay*. The left screen shows the hyperspectral image which is the content of BIL-formatted hyperspectral file. The image can be rotated by 90, 180, or 270 degrees. The right screen shows amplitude versus bands plot corresponding to a pixel located in the right screen. This plot can be exported to a spreadsheet file. In our case, the exported plot is stored in Excel format (.xls). The two fields that will be stored in the Excel file are the wavelengths and intensity of the corresponding wavelength associated with that pixel. The content of this file can be accessed later using Application 2 itself or using a regular spreadsheet application like Excel. Again, the users can activate the left screen and select a pixel of the image using the pick-up tool at the bottom-left corner of this screen. The information of the pixel will be shown in the left screen in a waveform plot.

The experiment can be stopped immediately by hitting the **Stop** button located at the top left of all three front panels. To start-up these applications, the control panels and graphs must be connected in the proper form in the block diagram. These diagrams will be explained in more detail in Section 4.2.

The list of configuration parameters in all three LabVIEW front panels is summarized in Table 4.1.1.
Table 4.1.1: Configurative Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integration Time (us)</td>
<td>Displays the actual integration time in microseconds</td>
</tr>
<tr>
<td>Spectral Binning</td>
<td>Combines spectral pixels</td>
</tr>
<tr>
<td>Move Absolute</td>
<td>Absolute stage movement to Target Position</td>
</tr>
<tr>
<td>% Scan Complete</td>
<td>Scan Progress Indicator</td>
</tr>
<tr>
<td>STOP</td>
<td>Stops Scan (Hold down for ~ 1 second to complete STOP)</td>
</tr>
<tr>
<td>Start Position (counts)</td>
<td>Start position of scan. 2000 counts = 1 mm</td>
</tr>
<tr>
<td>Scan Length (mm)</td>
<td>Total scan length</td>
</tr>
<tr>
<td>Step Size (um)</td>
<td>Distance stage travels between image captures (step size)</td>
</tr>
<tr>
<td>Create Envi File From Scan</td>
<td>Saves data to user defined band interleaved (BIL) data file. No extension</td>
</tr>
<tr>
<td></td>
<td>added to file name. An ENVI compatible header file is created with the same</td>
</tr>
<tr>
<td></td>
<td>name and a .txt extension automatically.</td>
</tr>
<tr>
<td>Start Acquisition</td>
<td>Start scanning</td>
</tr>
<tr>
<td>IMAGE</td>
<td>8-bit grayscale or 24-bit color (RGB) image of data. Data plotted in original</td>
</tr>
<tr>
<td></td>
<td>bit depth.</td>
</tr>
<tr>
<td>GRAY</td>
<td>Generates 8-bit grayscale image of the scanned object at user selected band</td>
</tr>
<tr>
<td></td>
<td>(wavelength). Bands loaded from header file.</td>
</tr>
<tr>
<td>RGB</td>
<td>Used only for visible spectrum scans. Generates color image of the scanned</td>
</tr>
<tr>
<td></td>
<td>object. Bands loaded from header file. Select bands closest to:</td>
</tr>
<tr>
<td></td>
<td>RED: 660 nm</td>
</tr>
<tr>
<td></td>
<td>GREEN: 567 nm</td>
</tr>
<tr>
<td></td>
<td>BLUE:: 460 nm</td>
</tr>
<tr>
<td>Scan Through Bands</td>
<td>Scans through grayscale images of object by changing band used to create</td>
</tr>
<tr>
<td></td>
<td>image</td>
</tr>
<tr>
<td>Scan Band Step</td>
<td>Sets “SCAN THROUGH BAND” step. Speeds up scan for large files</td>
</tr>
<tr>
<td>Scan Delay</td>
<td>Sets “SCAN THROUGH BAND” time delay (ms). Slows scan for smaller files.</td>
</tr>
<tr>
<td></td>
<td>Allowable values 0 to 10,000 ms</td>
</tr>
</tbody>
</table>

4.2 The Graphical Program

The graphical diagrams are the main part of the LabVIEW applications. Connections between the virtual instruments (VIs) and analog I/O are implemented under LabVIEW 2010 using the G programming language. In Figures 4.2.1-4.2.3, the graphical diagrams of the Start-up Page of Figure 4.1.1, “Application 1” of Figures 4.1.2 + 4.1.3,
and “Application 2” of Figure 4.1.4 are shown, respectively. The full G code of these programs can be found in Appendix A.

![Graphical Diagram of the Start-up Program](image)

**Figure 4.2.1: Graphical Diagram of the Start-up Program**

In general, each graphical diagram is divided into modules corresponding to configurative parameters. Each module is called the appropriate subVIs serving as sub-functions. For example, the module named *Spectral binning*, which is to adjust the binning parameter of the camera, needs to call subVIs *GetBinning.vi, SetBinning.vi* and another related subVIs from LabVIEW library named *Sensor.llb*. In the figures, only main modules are shown.

In “Application 1” graphical diagram, the main modules are *Camera configuration, Stage controller configuration, Create & Store a BIL file and a header file under ENVI format, Show real-time camera’s screen, and Plot real-time hyperspectral data*. Names of these modules show the missions of Application 1: Configure the camera...
and stage controller’s parameters, and then scan a sample to get a hyperspectral image which contains two files, a data file (BIL file) and a header file (HDR file).

Similarly, in “Application 2” graphical diagram, the list of the main modules is: Viewing Configuration, Open and show the content of a hyperspectral image, Plot stored hyperspectral data, Save the plot to a spreadsheet, and Open and read the content of a spreadsheet file. Thus the jobs which Application 2 can do are: Open and show a hyperspectral image as well as configure the viewing mode; Plot stored hyperspectral data as well as write and read the plot to/from the spreadsheet.

Figure 4.2.2: Graphical Diagram of Application 1

Similarly, in “Application 2” graphical diagram, the list of the main modules is: Viewing Configuration, Open and show the content of a hyperspectral image, Plot stored hyperspectral data, Save the plot to a spreadsheet, and Open and read the content of a spreadsheet file. Thus the jobs which Application 2 can do are: Open and show a hyperspectral image as well as configure the viewing mode; Plot stored hyperspectral data as well as write and read the plot to/from the spreadsheet.
4.3 The Website

In this study, a Website is designed to perform experiments with Hyperspectral Imaging System (HIS) remotely. A LabVIEW Web server 2010 is built to support the Start-up page and two application pages described in the previous sections. This Web server operates dedicated port to LabVIEW HTTP request. An Apache Web server is installed in the same physical server, but operating in different port. This Web server serves the regular HTTP request coming from the Internet.

The Web site is designed in accordance with Microsoft Internet Explorer 7.0 or higher and Mozilla Firefox 5.0 or higher. In the real-time application, the Web site allows only one user to control the system at the same time because of sharing only one set of physical scanning devices. However, multiple users are accepted to view the system simultaneously without changing control parameters. When the user logs in to the system, he or she requests control to obtain mutually exclusive access to the system. If the system
is available, the central server gives the user the requested control. The control is granted to the user for a maximum of five minutes in case another user is waiting to use the system. If no one else is waiting for the HIS control, the current user can have unlimited time on the HIS. If the system is not available at the time of user request (if another user is using the system), this request is put into a queue. During this time, the candidate user can only see the activities which are being performed by the controlling user on his or her computer screen. The control will be granted to the user in line either when the user in control of the HIS closes the Web browser, or the maximum five minutes pass. The database of the users and related logging information is stored in a MySQL database.

National Instruments has developed a Dynamic Link Library (DLL) that comes with the NI Internet Toolkit and NI Web Builder. Through this DLL, the virtual instrument developed in LabVIEW can be integrated in .NET easily. Thus LabVIEW components such as the control panels, graphs or plots can be converted to corresponding components in .NET when we want to publish the LabVIEW front panel to the Web. Many functions and components presented in DLL have been used in our designed Web site.
In Figure 4.3.1, the user authentication page is shown. Similarly, the corresponding pages of the LabVIEW front panels are shown in Figures 4.3.2, 4.3.3 and 4.3.4, respectively.

Figure 4.3.1: Web-based User Interface of the Homepage

Figure 4.3.2: Web-based User Interface of the Start-up Page
Figure 4.3.3: Web-based User Interface of Application 1

Figure 4.3.4: Web-based User Interface of Application 2
The necessary steps for conducting and performing the experiment are given as follows:

- First, username and password for authentication is requested from the user in the Homepage (Figure 4.3.1).
- After a successful log on to the session, the user interface for starting the applications of the HIS will appear in the Start-up Page (Figure 4.3.2).
- The user clicks on the right-mouse button and chooses **Requests control** to be granted system control for the maximum amount of time for a single user. If the control is granted, the user can select either of the two applications based on what is to be accomplished in a particular working session. Once an application is chosen, the other application is automatically disabled. The user cannot start the two applications at the same time.
- In the selected application, the user now can change the experiment parameters such as **Start position** of the stage, **Scan length** of the image or **Integration time** of the camera. (Figures 4.3.3 and 4.3.4)
- Since there is only one physical HIS in the lab, the Web site allows access to only one user at a time. While one user is performing the experiment, other users may only observe the experiment simultaneously. During the session, if the user with active control changes the parameters of the experiment, the other users can only watch these changes from their screen.
Each user is allotted five minutes and the session can be terminated by the user any time or after the default time limit of five minutes. The message The computer [ ] is controlling the system. is shown on the Web page of users who are on the waiting list. The [ ] contains the IP address of the computer which is controlling the HIS. When the user finishes the experiment or the default time limit is over, the control is switched to a user who is on top of the waiting list. Priority is assigned on a first-come first-served basis.

In order to switch between the applications or disable/enable the system, the user needs to go back the Homepage and hit Refresh button. When the system is deactivated by the system manager using the central server in the laboratory, nobody can access the experiments. The central server now has the system control. When the system is deactivated, the other users can only observe the experiment which is performing by the system manager.

The website is now available at http://hopilab.tamucc.edu (working with Internet Explorer 7.0 and higher or with Mozilla Firefox 5.0 and higher browsers).
V. DATA IN THE REMOTE CONTROL HIS

So far the design of the remote control HIS has been described. This chapter focuses on data flow in the HIS. How are the data created? How many types of data exist in the system? Where are the data stored? How are these data types shared? And how are the hyperspectral data viewed by the system and by other software? All these questions are answered in detail consecutively.

5.1. Data Acquisition

This section describes how the camera captures/records hyperspectral images. After the spectrum appears in the spectrograph, the mission of the camera is to record spectral images one line at a time fast and combine these lines into a hyperspectral image at the end of the process. There are two modes to record spectral images from the camera. One is recorder mode and one is first in first out (FIFO) buffer mode. Differences between these two modes are described in Table 5.1.1.

<table>
<thead>
<tr>
<th>Table 5.1.1 Recorder Mode vs. FIFO Buffer Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Recorder mode</strong></td>
</tr>
<tr>
<td>• Images are recorded and stored within the internal camera memory (camRAM).</td>
</tr>
<tr>
<td>• “Live View” transfers the most recent image to the PC (for viewing / monitoring).</td>
</tr>
<tr>
<td>• Indexed or total image readout after the recording has been stopped.</td>
</tr>
</tbody>
</table>
The recorder mode is divided into two sub-modes: sequence and ring buffer. The main difference is when the camera will stop recording. With the sequence sub-mode, recording is stopped when the allocated buffer is full. With the ring buffer sub-mode, camera records continuously into ring buffer. If the allocated buffer is full, the older images are overwritten. Recording is stopped by software command. As mentioned, the size of combined hyperspectral image is huge, in the order of several gigabytes. Thus spectral images cannot be stored temporarily in camera’s buffer as in recorder mode. The solution is to use FIFO buffer mode to temporarily transfer all spectral images to the central server in chronological order. These spectral images will be deleted automatically by scanning software after all these images are combined. In this case the scanning software is the “Application 1” mentioned in Chapter 4. The FIFO buffer mode is slower than the recorder mode but the correction of images is ensured. A LabVIEW library has been developed to perform the recording job. Firstly, the camera needs to be initialized using the OpenCameraEx.vi. Information about the camera connected is obtained using GetCameraType.vi and GetDesc.vi. Camera parameters are set using the VI’s in the driver library. Parameters are uploaded to the camera using the ArmCamera.vi. Once the camera is armed, the image size can be queried using GetSizes.vi. The size of the array is then modified to fit the image. A buffer is allocated for viewing images while recording. Setting the recording state to “Run” with the SetRecordingState.vi starts the recording process. Figure 5.1.1 shows the graphical diagram of subVI which initializes the camera. The graphical diagram for recording is shown in Figure 5.1.2.
5.2. Data Storage

The hyperspectral data obtained from the remote HIS are classified into two types of files: the hyperspectral image (the cube) and hyperspectral spreadsheet files. Hyperspectral images include two files: a Band Interleaved by Line (.BIL) file and a
header (.HDR) file. The .BIL file is the main part of a hyperspectral image, which is an uncompressed file containing the actual pixel values of an image. Black and white, grayscale, pseudocolor, true color, and multi-spectral image data can be handled by BIL format. The BIL file stores pixel information in separate bands within the file, thus the user can pick just one specific band to display in a multi-band image. The HDR file is an image description file (ASCII text format) that accompanies the BIL data. Header file describes the nature of the image data using intuitive keywords and values. The example below describes the header file in detail. The remote HIS system at HOPI lab has a line scan camera which is able to scan/capture only one row of pixels in an image. This is like a normal camera but the image resolution in terms of the full picture is some X by 1. In this example it is 800x1. Actually the camera returns an 800x270 resolution image, but what the number 270 represents is the 270 energy bands that the camera has captured for that single row of 800 pixels. So the camera has captured 270 bands of information each of resolution 800x1. After this is, the stage moves by a row (very small) and the camera captures another 270 bands of information. Many bands will be captured in this manner. Assume that the camera has captured 270 bands of information in 300 different positions of the camera. For example, take an 800x270 image, move by a row and take another image, etc. 300 times. Now the ENVI header file should have the following information:

```
ENVI description = {ENVI File, Created [11/17/2011 3:36 PM]}

Samples = 1600
Lines = 300
Bands = 800
header offset = 0
```
file type = ENVI Standard

data type = 14

interleave = bil

sensor type = Unknown

byte order = 0

wavelength units = Unknown

Some of the important fields from the above list are explained below:

- **Samples** - represents the horizontal resolution of the image being captured (1600 in the 1600x800 resolution).

- **Lines** - represents the number of rows of the image you will be capturing that will then be combined to form the spatial-spectral image; therefore, lines refer to the number of lines (rows) that will be present in the final image that will be constructed with all the information captured.

- **Bands** - this represents the number of energy bands that the camera captures in each image, in this example this value is 800 (the second part of 1600x800).

- **header offset** - indicates whether there is any header information before each image information. Value of zero means the image data starts from the top of the file.

- **file type** - ENVI standard (it depends on the scanning software using)

- **data type** – refers to how many bits are taken to represent each pixel. In this example, the camera takes 14 bits to represent each pixel.
• Interleave - indicates how the data is organized in the binary file. In this case, it is BIL file.

As mentioned, a hyperspectral image actually contains two files, the header which has the above information and the second which is a binary file containing the image information described in the header file. Thus the header file is like a map of the image information. The data structure of the BIL file can be explained as follows: The first row in the binary file will have the first row of the spatial image slice corresponding to the first band in the hyperspectral image. The second row of the BIL file corresponds to the first row of the image slice associated with the second band. The third row of the BIL file represents the first row of the spatial image slice corresponding to the third band, and so on. For the above example, the 812th row in the BIL file will be the second row of image slice corresponding to the first band. The 1623rd row is the third row of the image slice corresponding to the first band.

Therefore 811 spectral images (number of bands) of spatial resolution 1600x300 are obtained. In other words, 811 image slices each with a spatial resolution of 1600x300 are acquired. Each of the final image slices represents one band worth of information. A row of pixels stitched together 300 times with 1600 fixed spatial resolution and 811 bands will form the final hyperspectral image cube. Based on this principle, the hyperspectral scanned images are created and stored in the remote HIS system. However this raises some questions: Where should the hyperspectral images be stored? Should the data be stored locally in the central server or remotely in a cluster of storage devices? The answers will be different for data viewing and data sharing. This difference is discussed in the next sections.
Hyperspectral spreadsheet file contains information about wavelengths and corresponding intensity of a particular pixel in the hyperspectral image. This file serves as an input of generating the waveform plot which shows the data in a user friendly way.

5.3. Data Viewing with the Web-based HIS

To scan a sample using the remote control HIS, users need to log into the system. The Website creates a session for logged-in user. After scanning the sample, the user may need to view the result in the same session. Then he/she will type file name of a hyperspectral image and run Application 2. In Application 2, content of hyperspectral image is shown in the left screen as in Figure 5.3.1. A particular pixel in the hyperspectral image can be picked using pick-up tool at the right bottom of the left screen. The information of the picked pixel is explained in the right screen under waveform format as in Figure 5.3.2.

![Figure 5.3.1: Left Screen of Application 2](image-url)
The information about wavelengths and intensity of the picked image can be exported to a spreadsheet. This spreadsheet can be loaded and viewed using Application 2 directly. Figure 5.3.3 shows the graphical user interface of the tool which is used to view the content of a spreadsheet online.
In this particular case of viewing the image, when the user is still in the same session and does not log out of the system, the hyperspectral image from the camera and the exported spreadsheet need to be stored in a temporary folder in the central server for faster allow a multi-functioning server. As discussed, the central server serves as the transmitting system command as well as offering the Web service. If this server is also used as a storage server, the overall performance of the system would decrease significantly. Moreover, the storage space of this server would be filled up quickly because of the huge size of hyperspectral images. However, in the case described here, the hyperspectral data are stored in the central server temporarily. These temporary data will be deleted when the user logs out of the system and his/her session is terminated. This will release disk space for the central server and avoids making the central server an unwanted storage server.

If the user logs out the system and then wants to come back to review his/her last result, the user will create a new session; however, the data were deleted right after the user’s last session had been terminated. This will be a big problem if the temporary data are not transferred and stored in a remote location rather than the central server. Relocating the data to a dedicated storage server like a FTP server helps the system...
performance increase as well as decrease the burden put on the central server. The data can be retrieved whenever user needs and can be shared with other authenticated users. Figure 5.3.4 is a graphical diagram which shows transferring hyperspectral data from the central server to the FTP server. This happens right before the temporary data are deleted from the system.

**Figure 5.3.4: Sending Hyperspectral Data Using FTP**

Application 1 offers a feature for users to view the real-time spectral images directly from the camera as long as the camera is available. The screen showing the real-time images is depicted in Figure 5.3.5. In this particular case, the spectral images are combined a hyperspectral image after these images are shown individually in the web-based user interface of Application 1. Thus these spectral images do not need to be sent to the central server as a huge hyperspectral image. They are loaded directly from the camera’s buffer and send to central server’s buffer one by one. The Web browser retrieves these images from the local buffer and translates to the Web language. Finally the users see the images in the Web form. The graphical diagram of subVI which gets spectral images from the camera and output to the screen is shown in Figure 5.3.6.
Figure 5.3.5: Real-time Camera Screen in Application 1

Figure 5.3.6: Graphical Diagram of Outputting Real-time Spectral Images
5.4. Data Sharing with the Web-based HIS

Data sharing is an essential demand many local-control laboratories have been pursuing and developing [Ferwerda 2006] and [Gomez 2001]. The amount of samples which need to be scanned by the HIS will increase quickly. Moreover there is no laboratory which has the ability to scan every sample. Thus integration of the HSI data is a new trend and is an effervescent topic in recent years [Govender 2007] and [Vladimir 2006]. In the HOPI lab at TAMUCC, the HIS data are stored in a cluster of FTP servers. The FTP is a network protocol which allows sharing the data efficiently. FTP applications always operates based on the Client-Server principle, thus programming for FTP applications must follow standards of Client-Server programming. In this kind of programming, a FTP server application is installed in a FTP server by opening specific ports. A FTP client application is installed in a workstation which is used to connect to the FTP server. After connection is established, users can download or upload the files from/to the server. The FTP servers which are being used in the HOPI lab have been installed the FileZilla FTP server. This open-source application allows laboratory’s administrator to manage FTP users efficiently. In the client side, a LabVIEW FTP client has been developed on PHP as shown in Figure 5.4.1. With this embedded FTP client, users do not have to install FTP client application in their own computer. They simply need to use a regular Web browser to connect to the FTP server. This kind of FTP service is called Web-based FTP.
To establish a working FTP session, users need to connect to the FTP server first by typing FTP username and password. If they do not have a FTP username and password, which must be created by laboratory’s administrator, they can log into the FTP server using an anonymous account. With the anonymous account, the users can only see and download the files in a public folder. If the user has an authenticated FTP account, he/she can see and download files in both public folder and his/her own private folder. The user can download the files from FTP server by referencing in a file list which shows every file this user can see. The FTP command GET will send the files to the FTP server and this server will return the files being called. Similarly, the users can upload their data.

Figure 5.4.1: Web-based FTP Client Authentication Page
to FTP server from their computers. To accomplish this, the user needs to first choose the files to be uploaded to the FTP server. The FTP command PUT will send the selected file to the FileZilla FTP server. The files will be saved into the remote user’s own private folder. The graphical diagram of the LabVIEW FTP client application is shown in Figure 5.4.2.

![FileZilla FTP client application](image)

**Figure 5.4.2: Web-based FTP Client Page after Authentication**
6. TESTING AND EVALUATION

This chapter describes the test plan for estimating the speed of FTP transmission for the remote-control HIS. The speed of FTP transmission should be within 15% of the provisioned speed. Otherwise the transmission is considered too slow. Some possible reasons for slow data transmission are listed below:

- Existence of other computers on the network.
- Network hardware failing.
- Physical trouble with the network line.

Because of the above limits, a testing lab was set up at an off-campus location to measure the FTP speed from this off-campus location to the FTP server in the HOPI laboratory. For accurate results, it is best to connect a single PC to the DSL modem with the red cross-over cable. The Internet line at the particular off-campus location was subscribed from the Time Warner Cable. The maximum download speed was 20 Mbps and the maximum upload speed was 5 Mbps. The PC was installed Windows 7 with 100/1000 Mbps NIC. The processor was 2.26 GHz dual core. The network configuration of the PC was set up to ensure correct internet settings.

To accomplish the test, the tester needs to type ftp and the corresponding IP address of the FTP server in the command line window as shown in Figure 6.1. Enter FTP Username and Password for the HOPI FTP server and then press Enter (Figure 6.1). After successfully logging into the FTP server, type bin and then press enter. Also type hash and then enter. Next the tester needs to download a file to determine the transfer rate that tester are receiving. This is done by typing get <filename> where the file name is referenced to a hyperspectral image from the HOPI FTP server. The size of the file
chosen here is about 1 GB. After typing in the filename, the tester will see a series of # signs scroll across the command line screen as shown in Figure 6.2.

Figure 6.1: FTP Server Command Line

Figure 6.2: Getting a File Using FTP Command
To determine the physical bandwidth, multiply the rate reported in kilo bytes per second, in this case 2176.94 KB/second (or KBps), by 8 to obtain the transfer rate in kilo bits per second (Kb/second or Kbps).

\[ 2176.94 \times 8 = 17415.52 \text{ (Kb/second)} \]  

(1)

This rate should be within 15% of the provisioned speed. In this case the provisioned bandwidth at author’s house is 20 Mb/sec = 20480 Kb/sec, thus the FTP transfer rate should not go under:

\[ 20480 \times 0.85 = 17408 \text{ (Kb/second)} \]  

(2)

Comparing the example presented in Eq. (1) and Eq. (2), in can be seen that the FTP transfer rate is higher than the minimum threshold (17415.52 > 17408). This process is iterated 10 times. The results are shown in Table 6.1. The results show that in 8 out of 10 tests, the transfer rate remains above the threshold speed of 17408 Kb/sec. Thus the FTP server is running efficiently in given conditions of the remote control HIS.

**Table 6.1: FTP Transfer Rate**

<table>
<thead>
<tr>
<th>Order</th>
<th>Time (s)</th>
<th>Bandwidth (KB/s)</th>
<th>Bandwidth (Kb/s)</th>
<th>&gt;17408 ?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>473.41</td>
<td>2203.73</td>
<td>17629.84</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>475.82</td>
<td>2192.6</td>
<td>17540.8</td>
<td>Yes</td>
</tr>
<tr>
<td>3</td>
<td>475.38</td>
<td>2194.6</td>
<td>17556.8</td>
<td>Yes</td>
</tr>
<tr>
<td>4</td>
<td>470.94</td>
<td>2215.31</td>
<td>17722.48</td>
<td>Yes</td>
</tr>
<tr>
<td>5</td>
<td>474.6</td>
<td>2200.7</td>
<td>17605.6</td>
<td>Yes</td>
</tr>
<tr>
<td>6</td>
<td>464.46</td>
<td>2246.2</td>
<td>17969.6</td>
<td>Yes</td>
</tr>
<tr>
<td>7</td>
<td>478.06</td>
<td>2182.28</td>
<td>17458.24</td>
<td>Yes</td>
</tr>
<tr>
<td>8</td>
<td>553.79</td>
<td>1883.87</td>
<td>15070.96</td>
<td>No</td>
</tr>
<tr>
<td>9</td>
<td>471.09</td>
<td>2214.57</td>
<td>17716.56</td>
<td>Yes</td>
</tr>
<tr>
<td>10</td>
<td>501.35</td>
<td>2080.93</td>
<td>16647.44</td>
<td>No</td>
</tr>
</tbody>
</table>
7. FUTURE WORK AND CONCLUSIONS

7.1 Future Work

Although the user management module has been developed, the database management was beyond the scope of this project. So far the authentication page uses user information from a MySQL database in the central server. As mentioned, this puts much burden on the central server which has to perform a many other jobs. Moreover, a different user database has been installed in FileZilla FTP server. This causes simultaneous existence of two user-database systems. Future work, therefore, involves relocating the MySQL database to the FTP server and combining the two database systems into one.

In addition, the security issues listed below need to be solved:

- First, although the information of users in the MySQL database is encrypted by SHA/MD5 hash algorithm, information of the user in the FTP database is not. A solution which can be used is to combine the FTP user database and MySQL databases. This leads to a better situation such that the FTP user database is also encrypted by SHA/MD5.

- Second, the transmitted data is not encrypted over the Internet. This leads potentially confidential information being extracted during transmission using techniques of man-in-the-middle attack. A solution can be using a more secure FTP protocol such as SFTP, FTPS, or FTP over SSH according to FTP transactions and using HTTPS according to HTTP transactions. These protocols will encrypt the transmitted data and prevent the man-in-the-middle attack.
• Third, using the same local network with the University will lead to unpredictable risks that come from inside the organization.

• So far the commands are sent back and forth from hardware devices to the central server in a serial manner. Image analysis can be faster if algorithms are applied to process control signals in parallel.

### 7.2 Conclusions

In this document, a seven-module remote-control HIS model that covers the HSI process from hyperspectral data acquisition, data analysis, data viewing, data sharing, data storage to user management and system security is proposed. Based on this model, an application interfacing a lab-based HIS with the Internet through NI LabVIEW Web server 2010, NI Internet Toolkit 2010 and related programs is implemented in the HOPI laboratory at TAMUCC. Using this LabVIEW application, the system can be controlled remotely; the data can be viewed real-time and on-line by individuals who are part of the research team but who are unable to access the lab physically. This project involves the development of the LabVIEW interface to achieve real-time remote control of the HIS, and management of the acquired data as well as efforts to secure the HIS system. The successful implementation of the remote HIS will benefit the collaborators who want to remotely contribute their data and analysis to the project. Additionally, burden of learning the system put on the new researchers is reduced significantly.
BIBLIOGRAPHY AND REFERENCES


APPENDIX A – GRAPHICAL DIAGRAM

This section shows main graphic diagrams of the system.

The graphical diagram of startserver.vi:
The graphical diagram of scan.vi:
The graphical diagram of showcube.vi: