Web-Based Test Bed for Fingerprint Image Comparison

Hany H. Ammar, Xuhui Zhen,
Diaa Nassar, Yingzi Jin
Dept. of Computer Science and Electrical Engineering
West Viginia University
Morgantown, WV, 26506-6109
hammar@wvu.edu

Abstract

This paper addresses the problem of testing an automated fingerprint comparison system.
In this research, a web-based test bed is developed to assist the experiments in Fingerprint Image Comparison (FIC). The FIC function, which supports the comparison between submitted fingerprints and candidate fingerprint, is one of the specified functions of the US Federal Bureau of Investigation (FBI) Integrated Automated Fingerprint Identification System (IAFIS). The FIC test bed provides an integrated environment to simplify the testing process and collects information from test results. A 3-tiered architecture is used in the test bed design, which is composed of test bed client, test bed server and fingerprint image database. These components can be distributed on the network and run on different platforms. Because the test bed client is designed as a Java Applet, the FIC experiment can be conducted in a Java-enabled Internet web browser. In this environment, a powerful computer can be employed remotely to handle the requests for computation and a web browser can be used to submit requests for performing specified experiments over the network.

Key word: Internet architectures, Web-based Testing Environments, Client-Server Java Environments, Fingerprint Image Comparison (FIC), Integrated Automated Fingerprint Identification System (IAFIS)

1. Introduction

Fingerprint Image Comparison (FIC), which supports the comparison between submitted fingerprints and candidate fingerprint, is one of the specified functions in the FBI Integrated Automated Fingerprint Identification System(IAFIS). The FIC system has been developed by WVU and each component does a specific part of job in a FIC stage. Current FIC components are developed by different people and in different periods of time. Although each component of the FIC system works well, some difficulties still make doing test inconvenient and tedious.

- There is no graphics user interface for the FIC system.
- FIC system currently does not have a build-in facility to select fingerprint image from database directly.
- There is no visualized presentation for people to see how each step in the current FIC system is proceeding.
- Fingerprint image database takes too much space on the workstation doing FIC experiments.
- Looking for fingerprint images from the NIST database is difficult.

In this research, a 3-tiered web-based test bed is developed to solve these problems to assist the experiment of FIC.

2. Three-Tiered Web-Based Test Bed

The key characteristic of 3-Tiered client-server architecture is the separation of a distributed computing environment into presentation, functionality, and data components. The architecture is shown in Fig.1 The 3-tiers are separated so that the components could be replaced by a different implementation without effecting the other tiers.

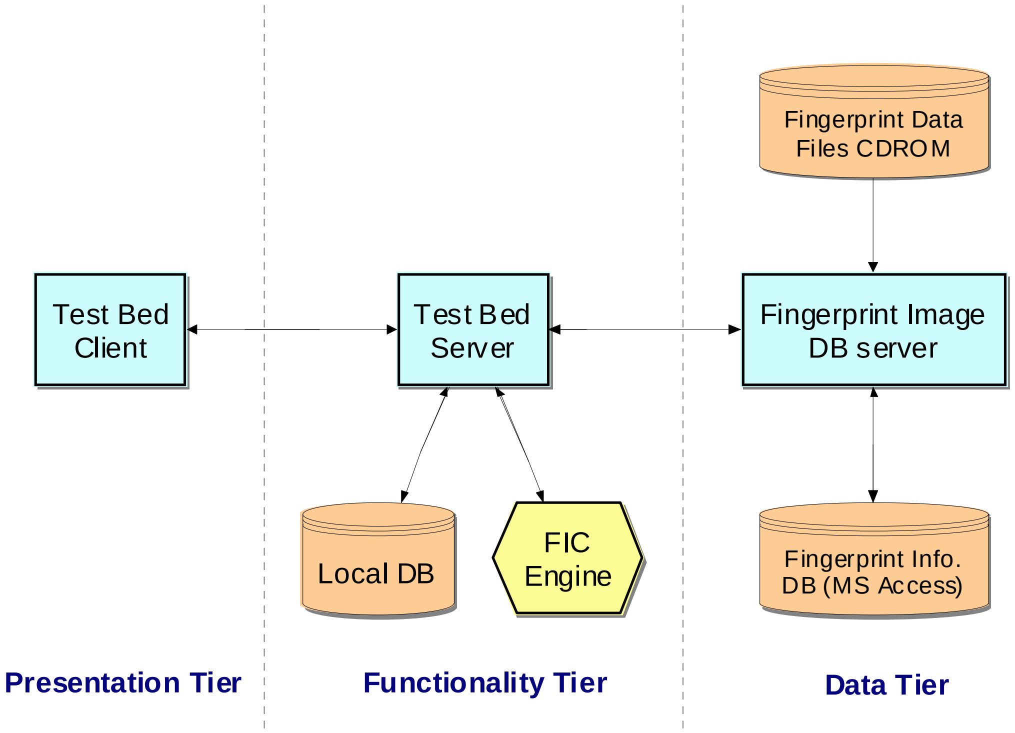

Fig.1 3-Tiered Architecture in Test Bed Design

- Presentation tier (Test bed client): It's the user interface; all the operations are performed from here. This is a place to submit user's testing samples and trigger the specific experiment on the test bed server. It is designed as a Java Applet, so that the FIC experiment can be conducted in a Java-enabled Internet web browser.
- Functionality tire (Test bed server): This tier does the real work for the FIC system. After receiving commands from clients, it can invoke the FIC engine and pass the results back to clients. Also, it is responsible for accessing the fingerprint database serv.
- Data tier (fingerprint image database server): It consists of the data files on the NIST CD-ROM and a relational database system for making query. It also provides service for database query and data file downloading. The attributes of the fingerprint images are stored in Microsoft Access database, and the image files themselves are kept on the CD-ROM. The whole database is separated from the test bed server so that less space is needed for the computer running the test bed server and the FIC engine. The FIC engine only works on the fingerprint image files in the local database, and other image files can be selected and downloaded from the database server to the test bed server's local database when needed.

Each tier communicates with other tiers using TCP connection, and commands and data are passed between tiers. User interacts at presentation tier; requests are generated based on the user's actions and sent to the functionality tier. Functionality tier either conducts a FIC task or forwards request to the data tier. The query results from database server, or testing results generated by FIC system on the test bed server, will be sent back to the client end. The details about test bed client, test bed server and fingerprint image database are described in the following sections.

3. Development of the Test Bed Client

The main task of the test bed client is to provide a friendly user interface. Commands could be generated as user requests and be sent to the test bed server. The responds could be sent back to the client after the test bed server finishing processing the requests sent from user.

The FIC system currently provides four functions, fingerprint image enhancement, extraction of the region of interests (ROI), alignment and neural network for decision-making. These are performed by the following client side functions (see Fig.2):

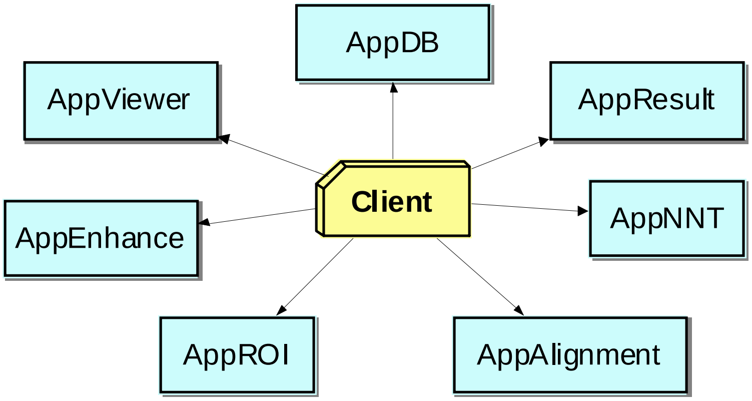

Fig. 2 Test Bed Client Design

- *AppViewer* is used to browse fingerprint image files on the workstation.
- *AppEnhance* is used to enhance image quality.
- *AppROI* is used to extract the regions of interests.
- *AppAlignment* is designed for performing alignment experiment.
- *AppNNT* is the interface designed for decision-making stage.
- *AppResult* is used to check previous testing results.
- *AppDB* provides a user interface for accessing fingerprint image database server.

Each client component has its own interface and can communicate with the server. To implement these client side functions, Java socket is used to perform network communication, threads are used for receiving multiple requests and dispatching requests to the right module. To make the communication between client and test bed server simplified, a set of formatted command is involved. The format is defined in Fig.3

cmd1	**cmd2**	**para**	**type**	**len**	**data[]**

Fig.3 Command Format

- *cmd1*: destination ID
- *cmd2*: source ID
- *para*: parameters, such as command id
- *type*: integer, byte, string
- *len*: size of data array *data[]*
- *data[]*: data array

There are two different image formats being used in this test bed, *PCT* and *RAS*. They have different file header format representing their features, for example, image size. The *PCT* file, in NIST CD-ROM, is compressed.

Java itself does not support those image formats, therefore, we first decompress the image files on the test bed server, get the image size from the file header. Then the file header of each image file is removed, only the pixel data and image size will be transmitted over network. The original image can be build from this raw pixel data by using *MemoryImageSource* class in Java.

The result from the alignment program includes a sequence of x offset, y offset and angle value, by which the image animation is built in the alignment process. We have another thread which keep calling the animation code to build different frame of images in order to achieve the animation effect.

While the program renders one image on the screen, it can build the next image in an off-screen buffer. When it is time for that next image to be displayed, it can be placed on the screen smoothly, the fading or chunking of the image will be hidden from us. In this way, the flicker is reduced tremendously. Fig.4 is a snapshot of the alignment modules in the test bed client.

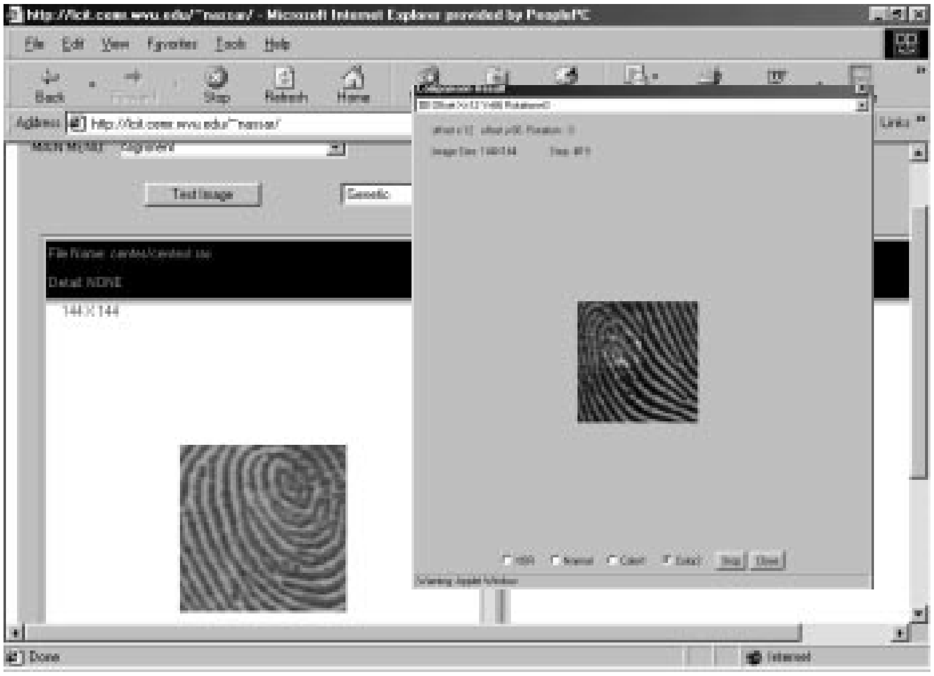

Fig.9 A Snapshot of Alignment on test bed client

4. Development of the Test Bed Server

When user interacts with test bed client, requests will be sent to the test bed server to complete specific jobs. Test bed server works as a middle ware for the test bed client to access fingerprint database server.

The test bed server have the following functions:

- provides services for many clients simultaneously.
- be able to pass parameters from the test bed client to the FIC engine and collect results when processing finished. The collected results will be sent back to client.
- The test bed server communicates with test bed client and fingerprint database server using TCP connection.

The test bed server is designed to be a multi-threaded application and each server thread is composed of five subsystems. The structure of test bed server is shown in Fig.5

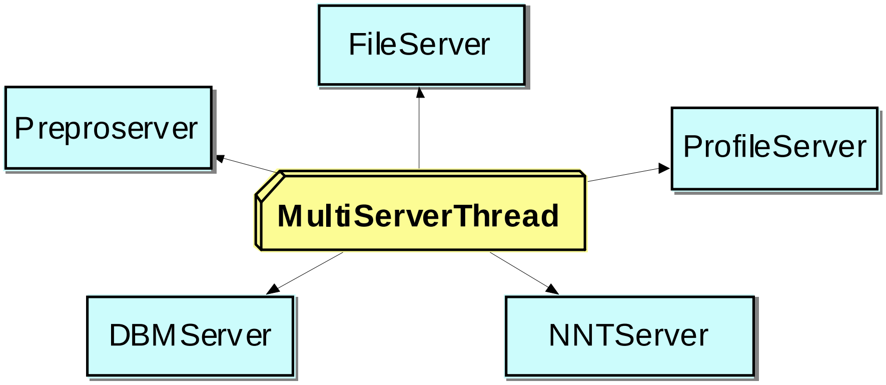

Fig.5 Test Bed Server Design

- The server thread provides a communication framework for all its components.
- *FileServer* is used to access local file system on the server and decompress fingerprint image data. It returns decompressed bit map image data to the client, then client can browse the fingerprint image on the workstation.
- *PreproSrever* is responsible for the preprocessing stage, including image enhancement, ROI extraction and alignment. The enhanced fingerprint image and ROI will be shown to the user, and temporary results are saved on the test bed server in order to provide inputs for the next step. A sequence of offset x, y and degree of rotation will be generated after alignment, this data will help the client animate how genetic algorithm performs for alignment.
- *DBMServer* takes care of the communication with database server. when test bed server received a command from client to access finger print image database server, *DBMServer* will make a TCP connection with the database server and exchange information.
- *NNTServer* pass parameters to the neural network based decision-making stage.
- *ProgfileServer* is used to store testing results and check previous experiments.

The test bed is implemented as a Java application, but programs in the FIC engine is written in C and C++. So Java Native Interface (JNI) is used to access the native code of C and C++. The use of JNI makes this software more efficient than pure Java code.

5. Development of the Fingerprint Image Database

The fingerprint image database holds the data files and the corresponding properties for specific fingerprint. Searching for the desired fingerprint can be performed from the selected test samples, the corresponding data of each fingerprint image will be transferred from fingerprint image database to the test bed server for further processing.

In this tier, the following functions should be included:

- In order to browse test samples, we should be able to search the whole database to find the desired fingerprint image based on different attributes, such as classifications (Loop, Whirl and etc.), sex, rolling and finger.
- The fingerprint image can be downloaded from the database server to the test bed server, where the actual FIC test performs.

For each image in the NIST fingerprint database, the corresponding attributes are stored in the file header, and the only way to know such information of certain fingerprint is to analyze the file header. It is very time consuming if searching function is based on the fingerprint image files themselves. To solve this problem, a relational database system is used. Searching is performed by DBMS and the fingerprint data files are sitting on the CD-ROM. The query results will be mapped to the data on CD-ROM.

The database server uses TCP socket to communicate with test bed server. First, client sends query for searching database to the test bed server, then test bed server will establish a connection to the database server and talk with it. After the test bed server received the query results, it forwards the information to the client and waits for other client commands.

Fig.6 is the design structure. The DBMS used in this research is Microsoft Access. Java Database Connecting (JDBC) and Open Database Connectivity (ODBC) are used as bridge between database server and Access database. JDBC technology is an API that lets you access virtually any tabular data source from the Java programming language, even in a heterogeneous environment.

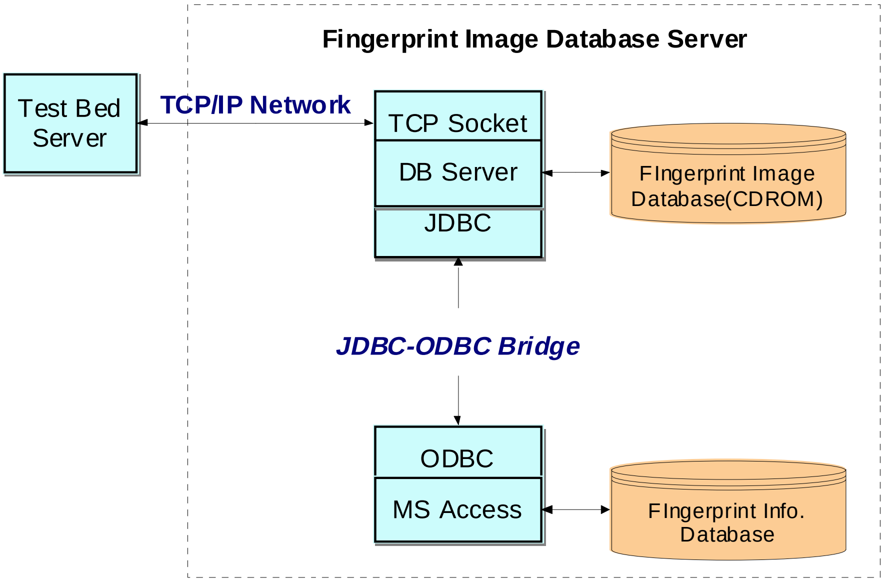
Fingerprint Image Database Server
Test Bed Server
TCP/IP Network
TCP Socket
DB Server
JDBC
FIngerprint Image Database(CDROM)
JDBC-ODBC Bridge
ODBC
MS Access
FIngerprint Info. Database

Fig.6 Design of Fingerprint Image Database Server

ATTRIBUTE	DATATYPE	CONSTRAINT
ID	Integer (Not Null)	Primary Key
Type	Integer	
Rolling	String	
Person	String	
Finger	String	
Class	String	
Sex	String	
Width	Integer	
Height	Integer	

Table 1 Fingerprint Table

There is only one table (Table 1), fingerprint, in the relational database. The attributes in this table are extracted form the information in NIST fingerprint database. Fingerprint database server is implemented as a multi-threaded daemon, which keeps listening on the network. Whenever a test bed server connects to the database server, a new thread will be created to handle this connection. The database server provides two services, query and download files. Query is performed by Access Database. The first step is to create the table (See Table1), and a program is written to extract information from each file header in the NIST fingerprint database and automatically build the Access database for the 4000 fingerprint images. The *path* attribute in the table is used for the system to find the corresponding data file from the CD-ROM, mapping between Access Database and Fingerprint image files on the CD-ROM.

6. Conclusion

In this research we presented a web-based test bed environment for a FIC system. It provides an integrated environment in which the testing process becomes simple and efficient. The Visualized process of fingerprint image comparison gives the FIC system developers more opportunities to evaluate the performance and effectiveness of the specific algorithm. Because this environment is totally web-based, we don't have to touch any other tools to complete a FIC experiment. The only thing needed to conduct FIC test is just a Java enabled web browser. 3-Tiered architecture gives us a distributed environment, data, user interface and the FIC system can be set up into different machines. In this environment, a fast machine can be use to handle the FIC tasks, a web browser can be used to control the tests over the network

7. References

[1] Xuhui Zhen.. "The Development of a Web-Based Test Bed For Fingerprint Image Comparison" MS CS degree problem report, Computer Science and Electrical Engineering Dept., West Virginia University, Morgantown, WV 25606, USA.

[2] Ammar, H.H., S. Zeng, and Z. Miao. "Parallel Processing and Fingerprint Image Comparison", Int. J. Modeling and Simulation, 18(2):85-99, 1998.

[3] Y. Tao 1999. "Fingerprint Registration Using Genetic Algorithms" MS CS degree problem report, Computer Science and Electrical Engineering Dept., West Virginia University, Morgantown, WV 25606, USA.

[4] Elliotte Rusty Harold, " Java Secrets", IDG Books Worldwide, Inc., ISBN: 0764580078 Pub Data: 05/01/97

[5] Jamie Javorski, "Java 1.1 Developer's Guide, Second Edition", Macmillan Computer Publishing, ISBN: 1575212838, Pub Date: 08/01/97

[6] Reese, G., "Database Programming with JDBC and Java", O'Reilly & Associates, Inc., 1997

[7]"The Java Native Interface Specification", http://java.sun.com/products/jdk/1.1/docs/guide/jni/spec/jniTOC.doc.html