Global Scientific JOURNALS

GSJ: Volume 8, Issue 6, June 2020, Online: ISSN 2320-9186
www.globalscientificjournal.com

BIOMETRIC AUTOMATIC ATTENDANCE SYSTEM IN NIGERIA HIGHER INSTITUTION

Salami Ibrahim Oluwatosin

Department of Electrical and Electronic Engineering,
Faculty of Engineering, Ekiti State University, Ado- Ekiti, Ekiti-State Nigeria.
Email: Ibrocology@gmail.com

ABSTRACT

This journal aims at introducing biometric capable technology for use in automating the entire attendance system for the students pursuing courses at an educational institute. The goal can be disintegrated into finer sub-targets; fingerprint capture and transfer, fingerprint image processing and wireless transfer of data in a server-client system. For each sub-task, various methods from literature are analyzed. From the study of the entire process, an integrated approach is proposed.

Biometrics based technologies are supposed to be very efficient personal identifiers as they can keep track of characteristics believed to be unique to each person. Among these technologies, Fingerprint recognition is universally applied. It extracts minutia- based features from scanned images of fingerprints made by the different ridges on the fingertips. The student attendance system is very relevant in an institute like ours since it aims at eliminating all the hassles of roll calling and malpractice and promises a full-proof as well as reliable technique of keeping records of student's attendance.

Keywords: Biometric Technology, Wireless Data Transfer, Arduino Mega 256, DSP starter kit TMS320C6713, Daughter card AFS8500/8600.

INTRODUCTION

The human body has the privilege of having features that are unique and exclusive to each individual. This exclusivity and unique characteristic has led to the field of biometrics and its application in ensuring security in various fields. Biometrics has gained popularity and has proved itself to be a reliable mode of ensuring privacy, maintaining security and identifying individuals. It has wide acceptance throughout the globe and now is being used at places like airports, hospitals, schools, colleges, corporate offices etc.

Biometrics is the very study of identifying a person by his/her physical traits that are inherent and unique to only the person concerned. Biometric measurement and assessment include fingerprint verification, iris recognition, palm geometry, face recognition etc. The above mentioned techniques work with different levels of functionality and accuracy.

Accuracy and reliability are the two most important parameters when it comes to biometric applications. Fingerprint verification is one of the oldest known biometric techniques known but still is the most widely used because of its simplicity and good levels of accuracy. It is a well-known fact that every human being is born with a different pattern on the fingers and this feature is exploited to identify and differentiate between two different persons.

The application in an educational institute is worth noting because of the benefits it brings along with it. The fingerprint recognition and verification technique can easily replace an attendance sheet and save time wasted on calling out roll numbers in the class. A fingerprint detecting device needs to be placed in each classroom and students would be made to swipe their finger over the sensor so as to mark their presence in the class. The database would contain all the fingerprints beforehand. So, the moment a finger would be swiped, a check would be carried out with the existing database and the corresponding student would get a present mark on his attendance record maintained in a server.

The transfer of the fingerprint from the device to the server can be carried out wirelessly using certain wireless adapters which can together form a wireless network in a short range and carry out the verification process. The communication channel needs to be secured and should be kept free from interference as far as possible. For further security of the entire system and to detect illegal activities, a security camera can be installed to keep track of the enrollments made in the classroom.

METHODOLOGY

Fingerprint identification is one of the most well-known and common biometric identification system. Because of their uniqueness and consistency over time, fingerprints have been used for identification over many years, more recently becoming automated due to advancement in computing capabilities. So, here the fingerprint identification technique was used for maintaining the attendance record. The record of the fingerprints of various students was maintained in a database. The communication between the PC and Module was done wirelessly over Bluetooth.

- For controlling both these modules the microcontroller board, Arduino Mega 2560 was used.
- For Implementing GUI Python's Tkinter library was used. The database was maintained over MySQL.

Students are supposed to enroll their fingerprint at the beginning of the semester for a particular course. During the class the fingerprint module would be passed among the students to mark their attendance.

Fingerprint Module GT-511C3

The module does all the heavy work of reading, identifying, and storing the fingerprint data. It can be issued several commands for all the functionalities. The module can store up to 200 different fingerprints and is capable of 360º recognition. For working the fingerprint must be registered by sending appropriate commands. On successful execution of the command it sends acknowledgement for success and Error code otherwise. The database of the prints can even be downloaded from the unit and distributed to other modules. The raw images of the fingerprints can also be retrieved from the module.

Enrolling

To enroll the fingerprint, the finger is to be pressed to the module thrice. All the three times it creates a template for the finger that was put on the optical sensor, added to that, the third time it also merges the three templates to create the final template. On successful enrolling the device sends a unique ID pertaining to the finger enrolled. This ID can be saved and later used for verification of the finger.

Enrolling Procedure

1. Enroll Start (ID); // Issue command to start enrolling over the past ID as parameter.
2. Capture Finger; // Take snapshot of the finger
3. Enroll1; // Create template of the 1st Image
4. Remove and press finger again
5. Capture Finger;
6. Enroll2; // Create template of the 2nd Image
7. Remove and press finger again
8. Capture Finger;
9. Enroll3; // Create template of the 3rd image and merge all 3 templates.

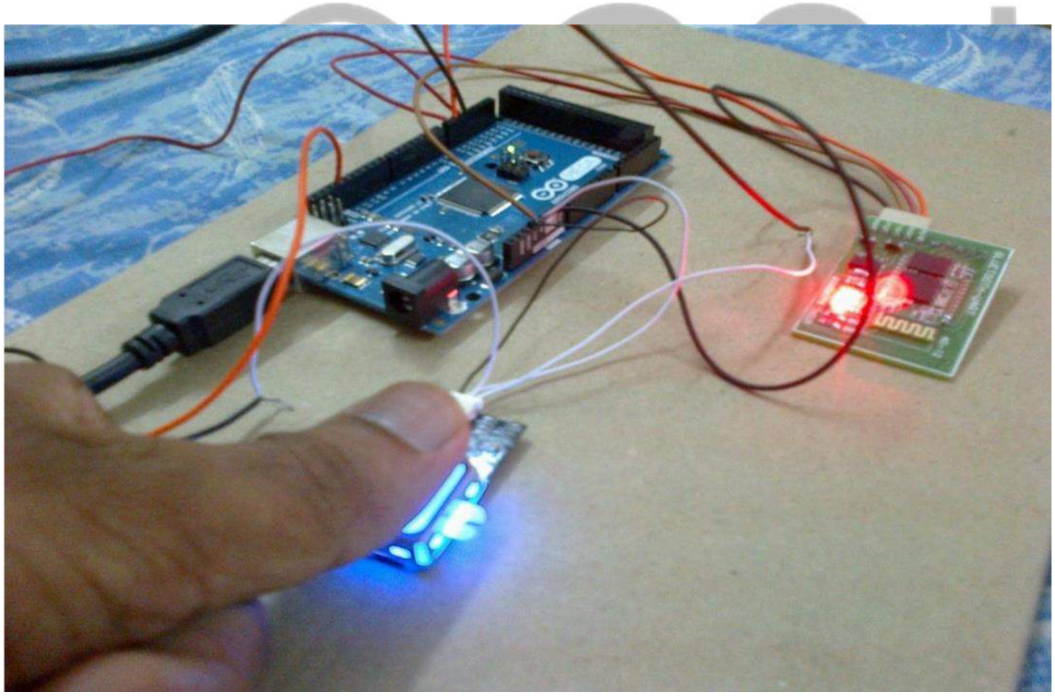


Figure 1: Fingerprint Module GT-511C3 Enrollment

Verification

Fingerprint verification is different from identification in a way that the person’s identity is stored along with the fingerprint in a database.

Verifying Procedure:

1. Capture Finger
2. Identify1_N
3. If (ID < 200) > Verified ID
4. Else Invalid Finger

During enrolling, we have to press the fingerprint module thrice, while during marking attendance, we have to press it only once.

Communication protocol

Connections: The fingerprint module has JST-SH connector with 4 pins. It needs a power supply of 3.3V-6V. But the Rx – Tx Communication voltage is 3.3V. So a potential divider is required to preventing the fingerprint module from the 5V Rx-Tx of the Arduino. The connections are as follows:

1. FPS Tx ----- Arduino Rx (Serial 2)

2. FPS Rx ----- Arduino Tx (Stepped down to 3.3V using potential divider of 560 and 1000 ohm resistors)

3. FPS Gnd ------Gnd

4. FPS VCC ------3.3V

The serial communication is set at 9600 bps on both the devices (Arduino and Fingerprint module) for correct synchronization. The fingerprint module has a default baud rate of 9600 bps.

CONSTRUCTION AND TESTING

Wireless Data Transfer

After the fingerprint image has been processed, the data is to be transferred to the central server through a wireless channel. The data packet is to be coded into an encrypted form due to the sensitive nature of the information it carries. The data communicated to the server is broadly classified into two types:

- Enroll Data
- Daily Attendance Data

Enroll Data

This data is initially obtained when adding the new students to the institute database. Along with Personal Identification Numbers (PIN), student-specific data such as degree program, date of birth (DOB), student picture and signature, the database is provided with a biometric template consisting of a processed image of the fingerprint.

Daily Attendance Data

Once all the students are enrolled into the institute's Student Attendance System, the daily work of each CHM is to accumulate the attendance data for each course for a particular classroom and transmit the data to the Central Server System (CSS). This data can be of two types:

Choice 1: Only the Status of Presence (SoP) of each student in the particular classroom is combined with his/her respective PIN (say Roll Number) into a Student Presence Data Packet (SPDP). Each SPDP is aggregated for the entire batch of students for the classroom and a Final Data Packet (FDP) is formed. This FDP is then transmitted to the CSS for each course class taken for that particular day.

Choice 2: The entire Fingerprint Template (FT) of each student present in that particular course class who performs a successful fingerprint capture at the CHM is combined with his/her respective PIN (say Roll Number) to form a SPDP. The SPDP of students present is accumulated into a FDP and this FDP is then transmitted to the CSS for that particular course class.

To decide about the choice of FDP from the above two options, we must look into the various pros and cons associated with each of them. Below is presented a comparative study of the various factors related to the above two choices.

Table 1: Final Data Packet Choice

FACTORS	CHOICE 1	CHOICE 2
Programming Complexity	More	Less
Data Packet Complexity	Less	More
Processing Time	More	Less
Time to Wait*	More	More

*Time to Wait (ToW): Time to Wait is defined as the time required before the CHM becomes ready to accept the next input fingerprint image through the Fingerprint Capture Module (FCM).

Clearly, we can see that the choice 2 option seems more appropriate. Regarding the data packet complexity, it is safe to assume that wireless channel remains relatively idle for the major part of the time and hence data can be transmitted from each individual CHM to CSS immediately, or by CSS defined rule. Either a timing based or a response-based rule may be used to accept data from each CHM.

On the CSS, the receiving wireless communication module (WCM) accepts the FDP from each CHM and converts it from electrical signals to digital data packets (DDP) which are then sent to the Server. The Server then parses each DDP, decomposes it into individual SPDP and then each SPDP into respective PIN and attendance data**. And determines the type of data the FDP contains; whether it carries an Enroll Data or Daily Attendance Data.

**The attendance data may be SoP for Choice 1 or FT for Choice 2.

If it finds that the received FDP contains Enroll Data, then it accesses the Fingerprint Database System, to add a new student to the institute database. If on the other hand it finds that the received FDP contains Daily Attendance Data, it may have to access both the Fingerprint Database and the Attendance Database. For the option of Choice 1, Attendance Database is updated directly with the latest attendance data using each individual SoP for that particular course class. For the option of Choice 2, the received FDP is decomposed into individual SPDP. Then each SPDP is decomposed into the respective PIN and its FT. First the Fingerprint Database is accessed using the respective PIN and then a Server-side matching of the two fingerprint templates is done. If match happens, the Attendance Database is updated. This step is performed for every DDP received.

EXPERIMENTAL SETUP

The actual testing for the design of the wireless fingerprint based student attendance system was carried out in Communications Lab., Department of Electrical Engineering. The experimental setup consists of both software based platform and hardware module in an integrated development environment. The various components of the testing environment are:

- TMS320C6713 DSK
- AFS8500/8600 (Daughter Card)
- Wireless G Desktop Adapter
- Code Composer Studio v2.0
- FRT in MATLAB

The individual components are illustrated in the subsequent pages in detail.

TMS320C6713 DSK

This is a SPECTRUM DIGITAL product that includes a Texas Instrument's DSP TMS320C6713 operating at 225 MHz, mounted over a DSP Starter Kit complete with JTAG emulation through on-board JTAG emulator with USB host interface or external emulator and a host of other features.

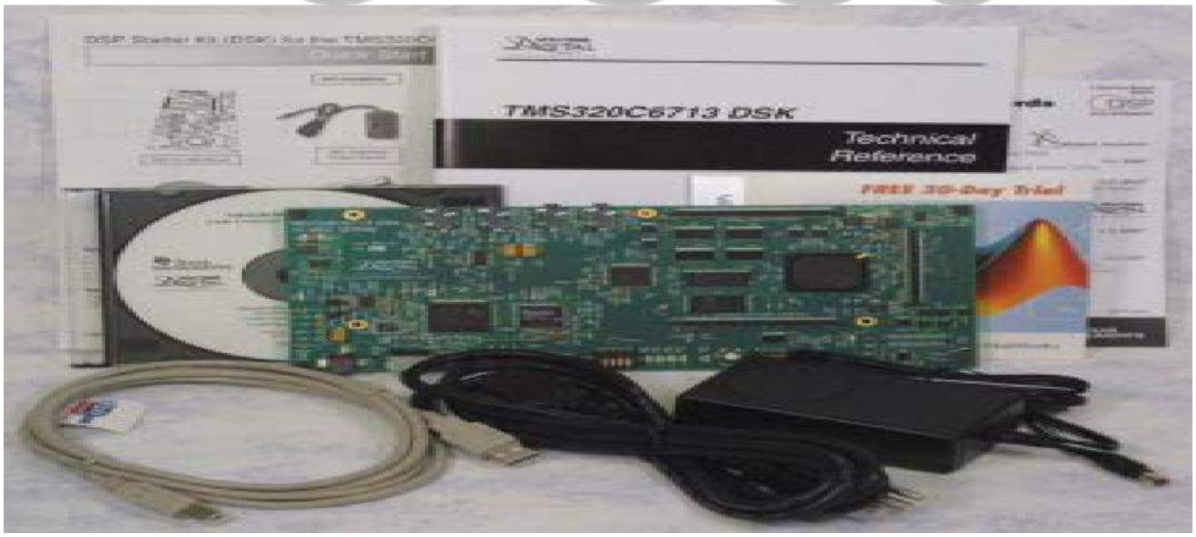


Figure 2: TMS320C6713 DSK

The TMS320C6713 DSK functions as the Processor Module with the option of either simply controlling the Fingerprint Module or along with control of Fingerprint Module, also carrying out the fingerprint image processing and creation of the Fingerprint Template.

AFS8500/8600 Daughter Card

It is a Texas Instruments product provided with an optical sensor for fingerprint image capture. It functions as the Fingerprint Capture Module.

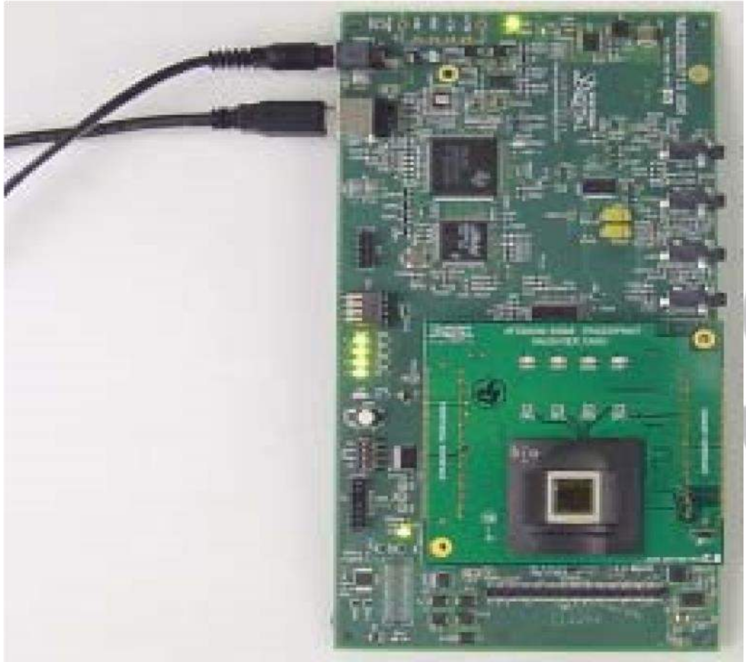


Figure 3: FDC-AFS8600 Sensor Board Mounted on C6713 DSK

Wireless G desktop adapter

It is known by its product name DWA-510, the D-Link Wireless G DWA-510. Desktop adapter features the very latest in advanced wireless silicon chip technology to deliver a maximum wireless signal rate of up to 54Mbps in the 2.4GHz frequency.

Some of its features are:

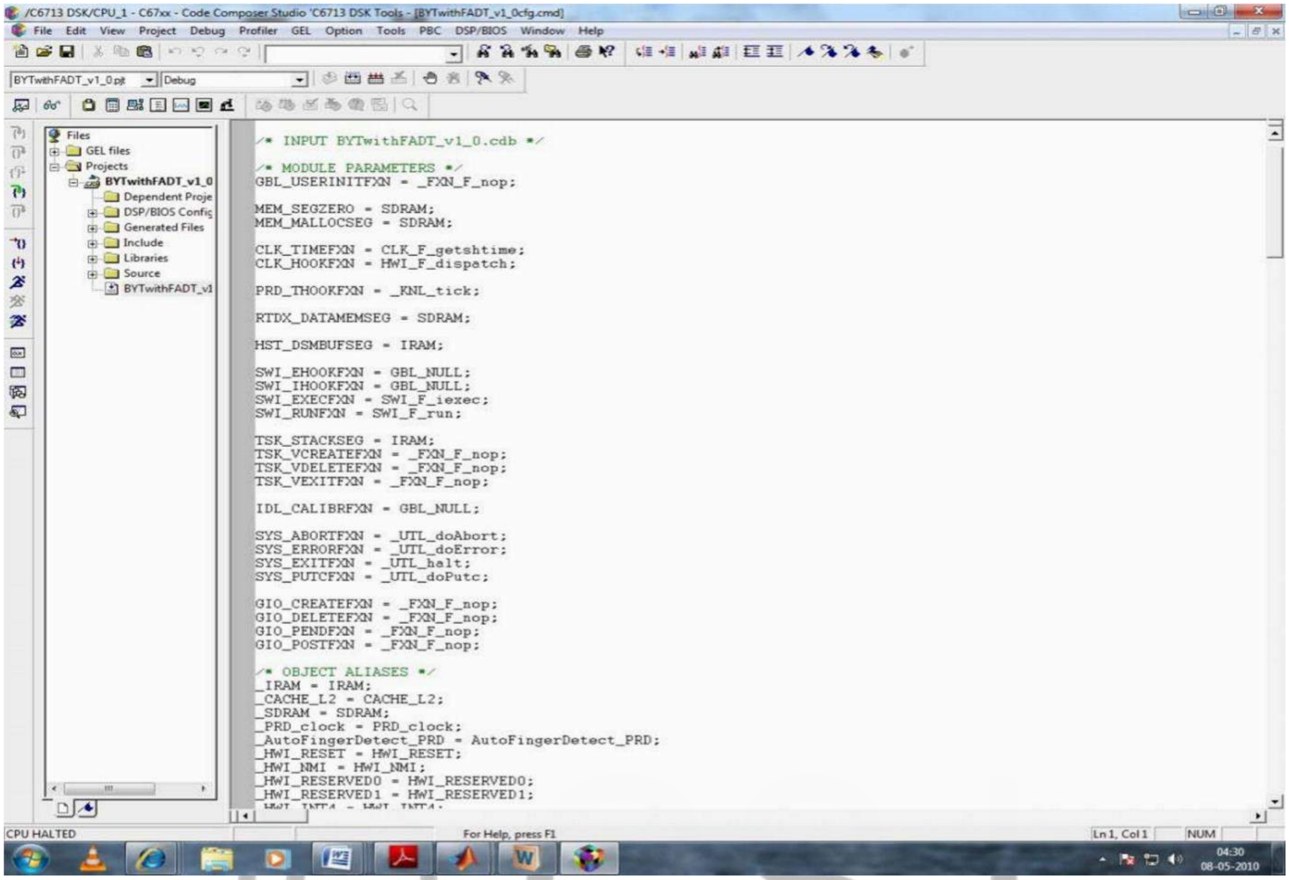
- Faster Wireless Networking.
- Compatible with 802.11b and 802.11g Devices
- 32-bit PCI Performance/Plug & Play Connectivity • User-friendly configuration and diagnostic utilities.

Figure 4: Wireless G DWA-510 Desktop Adapter

This functions as the wireless communication module for the purpose of data transfer between two PC. Security features such as WPA, WPA2, and WEP allow for secure and encrypted channel.

Code Composer Studio v2.0

Code Composer Studio (CCS) from Texas Instruments consists of a host of utilities that can be used for development and debugging of embedded applications. It provides a fast and comfortable interface to each step of code development. Special support for TI's devices such as compilers, source code editor, project build environment, debugger, profiler, simulators and many other features are included.



```
/* INPUT BYTwithFADT_v1_0.cdb */

/* MODULE PARAMETERS */
GBL_USERINITFXN = _FXN_F_nop;

MEM_SEGZERO = SDRAM;
MEM_MALLOCSEG = SDRAM;

CLK_TIMEFXN = CLK_F_getshtime;
CLK_HOOKFXN = HWI_F_dispatch;

PRD_THOOKFXN = _KNL_tick;

RTDX_DATAMEMSEG = SDRAM;

HST_DSMBUFSEG = IRAM;

SWI_EHOOKFXN = GBL_NULL;
SWI_IHOOKFXN = GBL_NULL;
SWI_EXECFXN = SWI_F_iexec;
SWI_RUNFXN = SWI_F_run;

TSK_STACKSEG = IRAM;
TSK_VCREATEFXN = _FXN_F_nop;
TSK_VDELETEFXN = _FXN_F_nop;
TSK_VEXITFXN = _FXN_F_nop;

IDL_CALIBRFXN = GBL_NULL;

SYS_ABORTFXN = _UTL_doAbort;
SYS_ERRORFXN = _UTL_doError;
SYS_EXITFXN = _UTL_halt;
SYS_PUTCFXN = _UTL_doPutc;

GIO_CREATEFXN = _FXN_F_nop;
GIO_DELETEFXN = _FXN_F_nop;
GIO_PENDFXN = _FXN_F_nop;
GIO_POSTFXN = _FXN_F_nop;

/* OBJECT ALIASES */
_IRAM = IRAM;
_CACHE_L2 = CACHE_L2;
_SDRAM = SDRAM;
_PRD_clock = PRD_clock;
_AutoFingerDetect_PRD = AutoFingerDetect_PRD;
_HWI_RESET = HWI_RESET;
_HWI_NMI = HWI_NMI;
_HWI_RESERVED0 = HWI_RESERVED0;
_HWI_RESERVED1 = HWI_RESERVED1;
```

Figure 5: CCS IDE

Fingerprint recognition toolbox

A new toolbox downloaded from the MATLAB CENTRAL website at www.mathworks.com allows us to add fingerprints to the database. Also it allows us to do a 1: n fingerprint match for verification.

It includes the various functions listed below:

- Fingerprint image visualization
- Gabor filter visualization
- Image enhancement
- Orientation field estimation • Core point localization

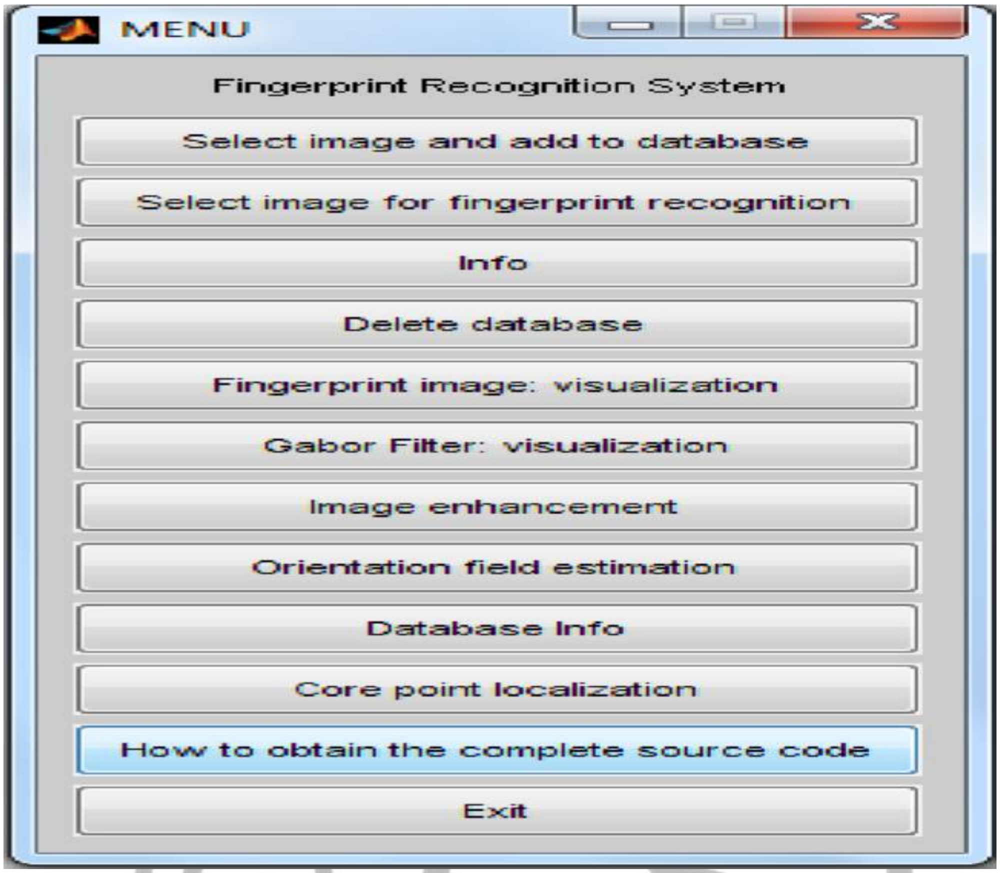

MENU
Fingerprint Recognition System
Select image and add to database
Select image for fingerprint recognition
Info
Delete database
Fingerprint image: visualization
Gabor Filter: visualization
Image enhancement
Orientation field estimation
Database Info
Core point localization
How to obtain the complete source code
Exit

Figure 6: FRT in MATLAB

RESULT

Initial progress is mentioned below:

i. The DSP starter kit TMS320C6713 and the Daughter card AFS8500/8600 were tested for proper functioning. The two were found to work properly.

ii. A demo software was run on the fingerprint module and its operation was analyzed. It was observed to be an Enroll-Once-Verify-Once software. The threshold for content matching was very low and flexibility for different orientations of the finger was not present.

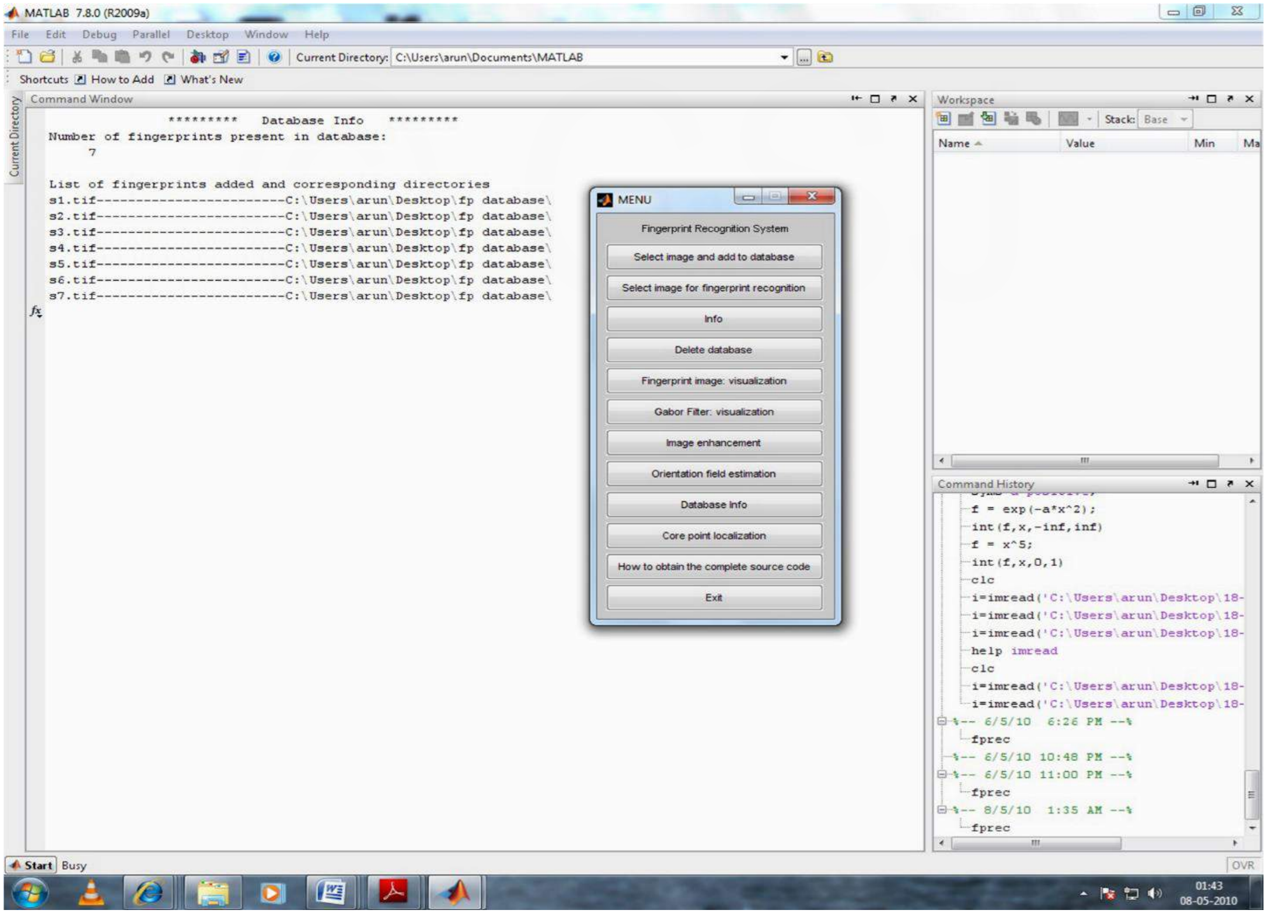
iii. Established wireless network involving two terminals using DWA-510.

The main objective of the project then was to enroll fingerprints of different students and add them to the database which would be referred at the time of verification. For this purpose, Fingerprint Recognition Toolbox provided for use in MATLAB was used. For a particular trial run of the system, fingerprints of eight students were captured using the hardware kit in the lab and fingerprint image of seven were added to the database. The templates stored were named from s1 to s7. To show the successful functioning of the system three sample outputs are provided that show;

Addition to database (result1)

A fingerprint match for s1 (result2)

A fingerprint match for s8 (result3)



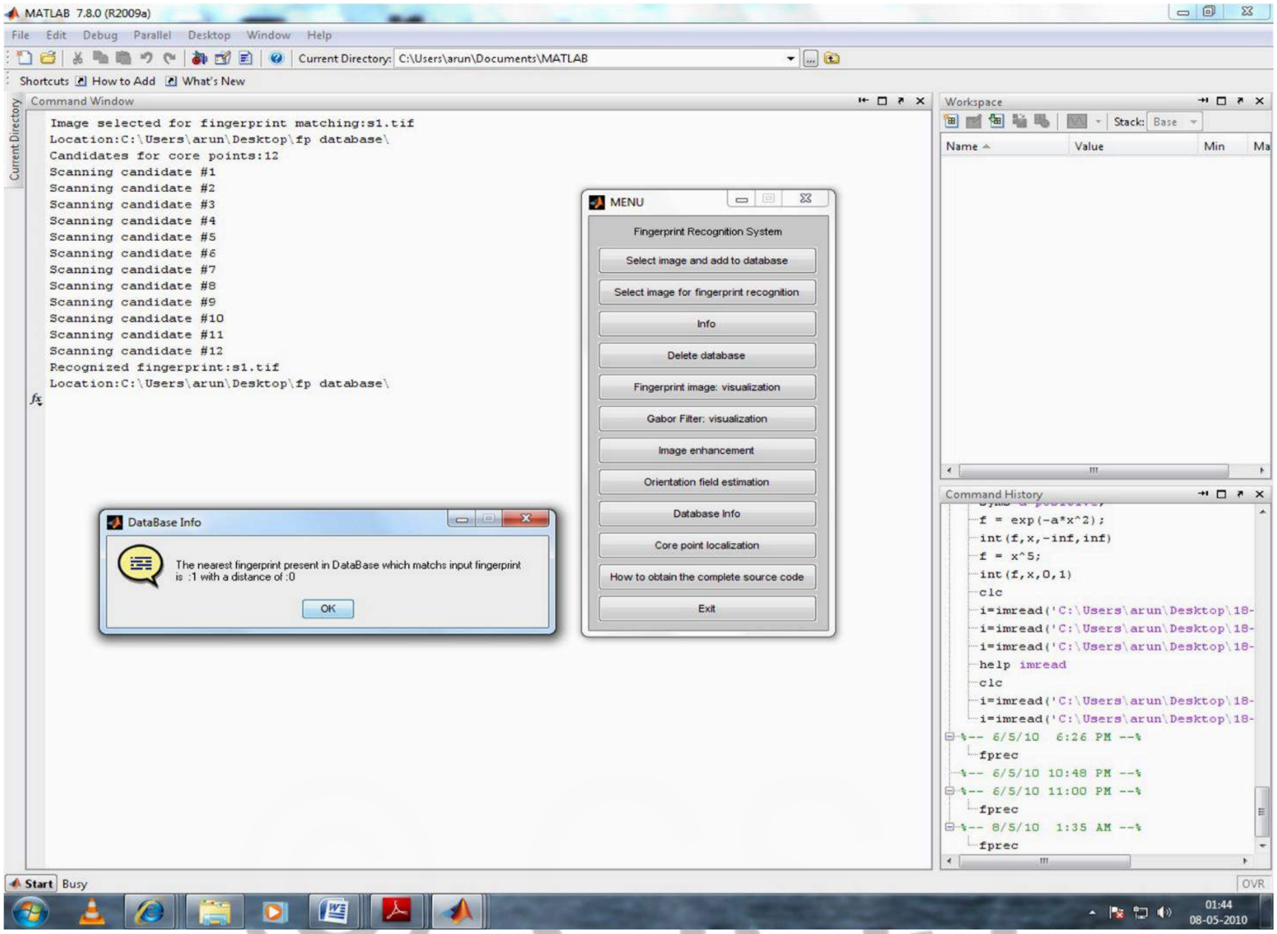
MATLAB 7.8.0 (R2009a)
File Edit Debug Parallel Desktop Window Help
Current Directory: C:\Users\arun\Documents\MATLAB
Shortcuts How to Add What's New
Current Directory
Command Window
Workspace
Stack: Base
Name Value Min Ma
MENU
Fingerprint Recognition System
Select image and add to database
Select image for fingerprint recognition
Info
Delete database
Fingerprint image: visualization
Gabor Filter: visualization
Image enhancement
Orientation field estimation
Database info
Core point localization
How to obtain the complete source code
Exit
Command History
Start Busy
OVR
01:43
08-05-2010

```
               *********   Database Info   *********
Number of fingerprints present in database:
     7

List of fingerprints added and corresponding directories
s1.tif------------------------C:\Users\arun\Desktop\fp database\
s2.tif------------------------C:\Users\arun\Desktop\fp database\
s3.tif------------------------C:\Users\arun\Desktop\fp database\
s4.tif------------------------C:\Users\arun\Desktop\fp database\
s5.tif------------------------C:\Users\arun\Desktop\fp database\
s6.tif------------------------C:\Users\arun\Desktop\fp database\
s7.tif------------------------C:\Users\arun\Desktop\fp database\
```

```
f = exp(-a*x^2);
int(f,x,-inf,inf)
f = x^5;
int(f,x,0,1)
clc
i=imread('C:\Users\arun\Desktop\18-
i=imread('C:\Users\arun\Desktop\18-
i=imread('C:\Users\arun\Desktop\18-
help imread
clc
i=imread('C:\Users\arun\Desktop\18-
i=imread('C:\Users\arun\Desktop\18-
%-- 6/5/10  6:26 PM --%
fprec
%-- 6/5/10 10:48 PM --%
%-- 6/5/10 11:00 PM --%
fprec
%-- 8/5/10  1:35 AM --%
fprec
```

Figure 7: Sample Matlab Output (Result1)

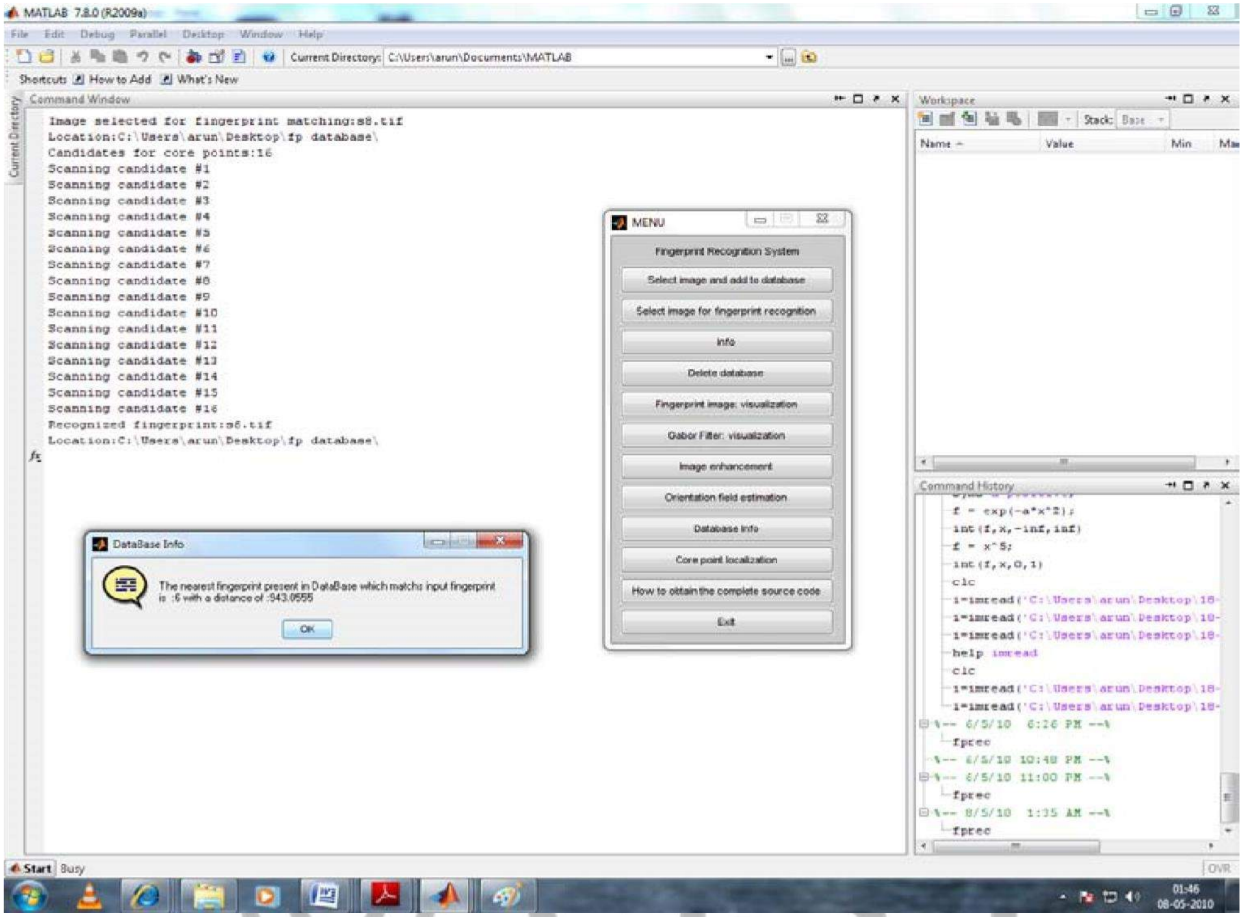

MATLAB 7.8.0 (R2009a)
File Edit Debug Parallel Desktop Window Help
Current Directory: C:\Users\arun\Documents\MATLAB
Shortcuts How to Add What's New
Current Directory
Command Window
Workspace
Stack: Base
Name Value Min Ma
MENU
Fingerprint Recognition System
Select image and add to database
Select image for fingerprint recognition
Info
Delete database
Fingerprint image: visualization
Gabor Filter: visualization
Image enhancement
Orientation field estimation
Database Info
Core point localization
How to obtain the complete source code
Exit
DataBase Info
OK
Command History
Start Busy
OVR
01:44
08-05-2010

```
Image selected for fingerprint matching:s1.tif
Location:C:\Users\arun\Desktop\fp database\
Candidates for core points:12
Scanning candidate #1
Scanning candidate #2
Scanning candidate #3
Scanning candidate #4
Scanning candidate #5
Scanning candidate #6
Scanning candidate #7
Scanning candidate #8
Scanning candidate #9
Scanning candidate #10
Scanning candidate #11
Scanning candidate #12
Recognized fingerprint:s1.tif
Location:C:\Users\arun\Desktop\fp database\
fx
```

The nearest fingerprint present in DataBase which matchs input fingerprint is :1 with a distance of :0

```
f = exp(-a*x^2);
int(f,x,-inf,inf)
f = x^5;
int(f,x,0,1)
clc
i=imread('C:\Users\arun\Desktop\18-
i=imread('C:\Users\arun\Desktop\18-
i=imread('C:\Users\arun\Desktop\18-
help imread
clc
i=imread('C:\Users\arun\Desktop\18-
i=imread('C:\Users\arun\Desktop\18-
%-- 6/5/10  6:26 PM --%
fprec
%-- 6/5/10 10:48 PM --%
%-- 6/5/10 11:00 PM --%
fprec
%-- 8/5/10  1:35 AM --%
fprec
```

Figure 8: Sample Matlab Output (Result2)


MATLAB 7.8.0 (R2009a)
File Edit Debug Parallel Desktop Window Help
Current Directory: C:\Users\arun\Documents\MATLAB
Shortcuts How to Add What's New
Command Window
Workspace
Stack: Base
Name Value Min Max
MENU
Fingerprint Recognition System
Select image and add to database
Select image for fingerprint recognition
Info
Delete database
Fingerprint image: visualization
Gabor Filter: visualization
Image enhancement
Orientation field estimation
Database info
Core point localization
How to obtain the complete source code
Exit
DataBase Info
OK
Command History
Start Busy
OVR
01:46
08-05-2010

```
Image selected for fingerprint matching:s8.tif
Location:C:\Users\arun\Desktop\fp database\
Candidates for core points:16
Scanning candidate #1
Scanning candidate #2
Scanning candidate #3
Scanning candidate #4
Scanning candidate #5
Scanning candidate #6
Scanning candidate #7
Scanning candidate #8
Scanning candidate #9
Scanning candidate #10
Scanning candidate #11
Scanning candidate #12
Scanning candidate #13
Scanning candidate #14
Scanning candidate #15
Scanning candidate #16
Recognized fingerprint:s6.tif
Location:C:\Users\arun\Desktop\fp database\
```

The nearest fingerprint present in DataBase which matchs input fingerprint is :6 with a distance of :943.0555

```
f = exp(-a*x^2);
int(f,x,-inf,inf)
f = x^5;
int(f,x,0,1)
clc
i=imread('C:\Users\arun\Desktop\18-
i=imread('C:\Users\arun\Desktop\18-
i=imread('C:\Users\arun\Desktop\18-
help imread
clc
i=imread('C:\Users\arun\Desktop\18-
i=imread('C:\Users\arun\Desktop\18-
%-- 6/5/10  6:26 PM --%
fprec
%-- 6/5/10 10:48 PM --%
%-- 6/5/10 11:00 PM --%
fprec
%-- 8/5/10  1:35 AM --%
fprec
```

Figure 9: Sample Matlab Output (Result3)

The various processes involved in the image processing of the captured fingerprint image using the FRT are explained below.

Fingerprint image visualization

It provides us with a visual picture of the fingerprint captured and transferred from the DSP TMS320C6713 to the server computer.

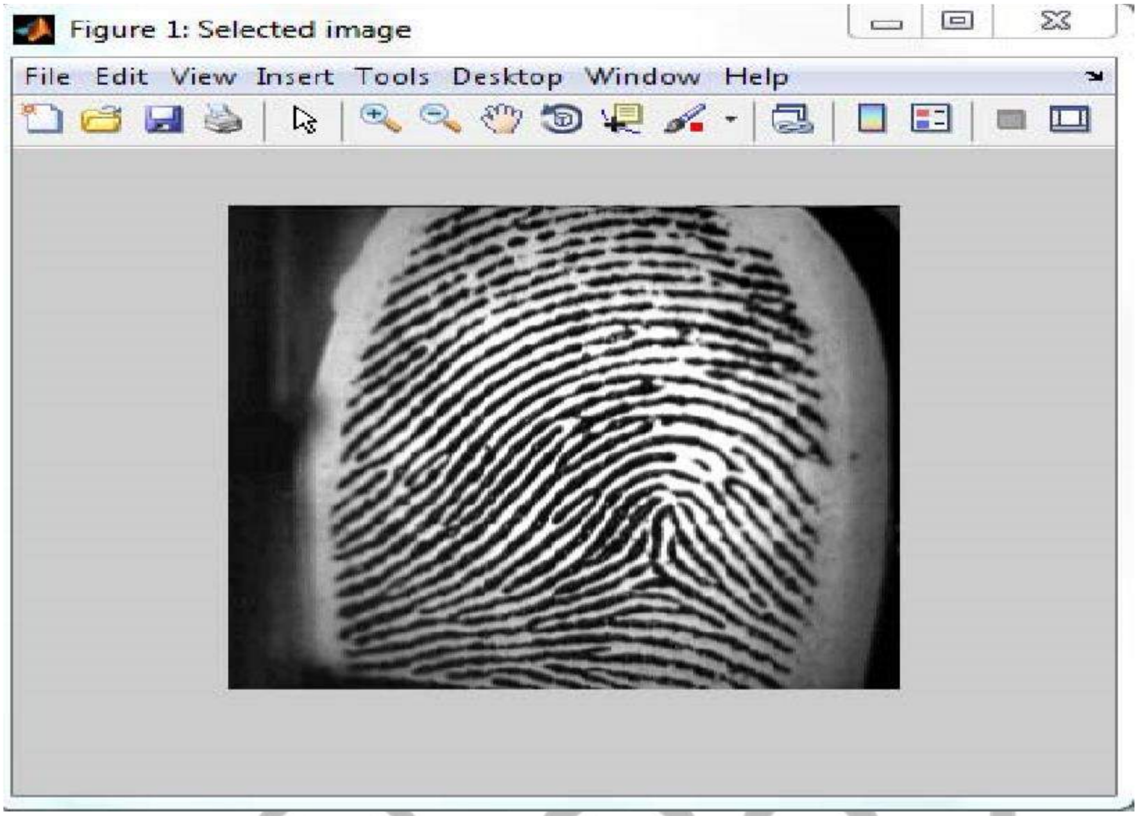


Figure 10: Fingerprint Image Visualization

Gabor filter visualization

A Gabor filter is a linear filter used in image processing for edge detection. Frequency and orientation representations of Gabor filter are similar to those of human visual system, and it has been found to be particularly appropriate for texture representation and discrimination. In the spatial domain, a 2D Gabor filter is a Gaussian kernel function modulated by a sinusoidal plane wave. The Gabor filters are self-similar - all filters can be generated from one mother wavelet by dilation and rotation.

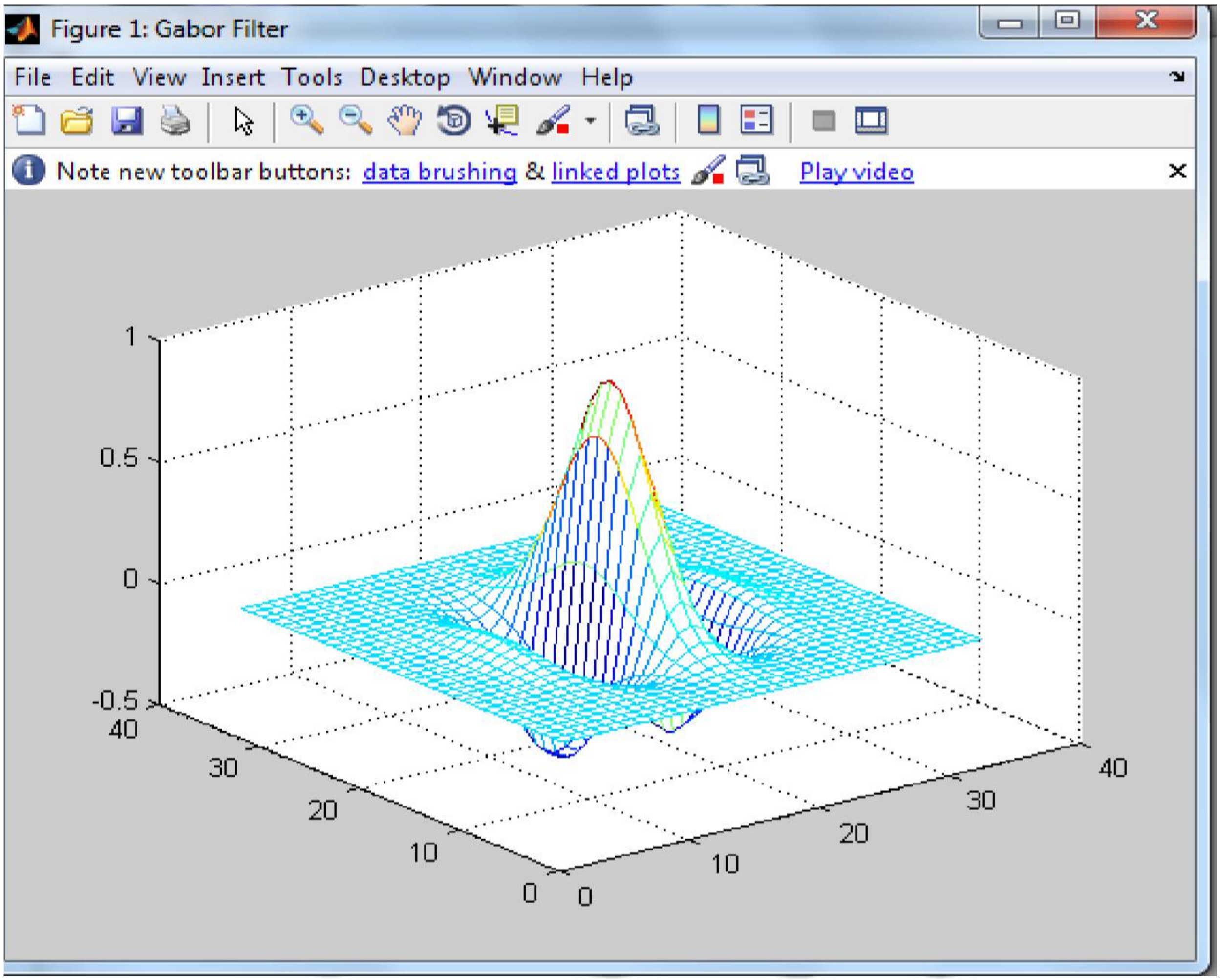


Figure 1: Gabor Filter
File Edit View Insert Tools Desktop Window Help
Note new toolbar buttons: data brushing & linked plots Play video
1
0.5
0
-0.5
40
30
20
10
0
0
10
20
30
40

Figure 11: Gabon Filter Visualization

Image enhancement

In order to ensure that the performance of an automatic fingerprint identification/verification system will be robust with respect to the quality of input fingerprint images, it is essential to incorporate a fingerprint enhancement algorithm in the minutiae extraction module. It adaptively improves the clarity of ridge and valley structures of input fingerprint images based on the estimated local ridge orientation and frequency.

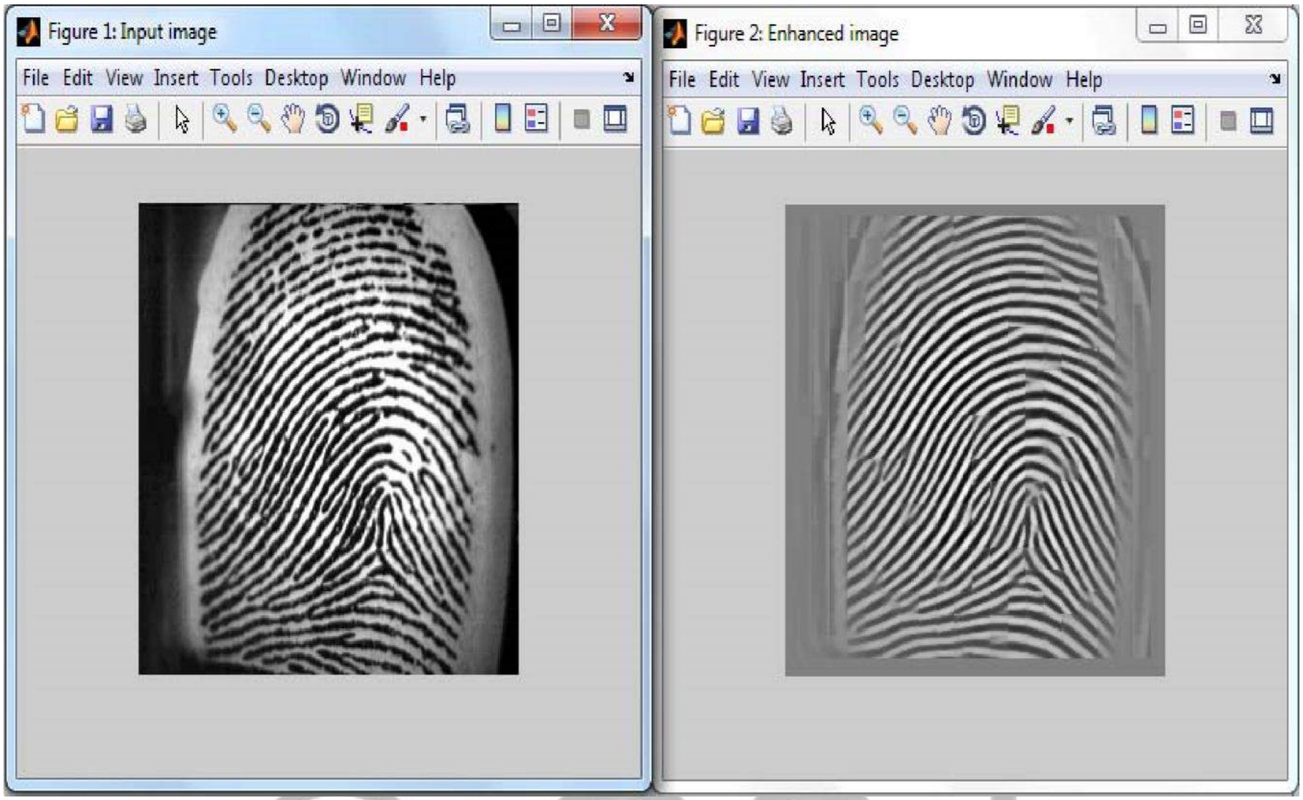


Figure 1: Input image
File Edit View Insert Tools Desktop Window Help
Figure 2: Enhanced image
File Edit View Insert Tools Desktop Window Help

Figure 12: Image Enhancement

As shown in the above picture, the image to the right is an enhanced version of the original input fingerprint which is on the left. The input image is segmented into a matrix of cells which are individually processed.

<u>Orientation field estimation</u>

A directional field describes the coarse structure of a fingerprint. It describes the local orientations of the ridge and valley structures, and is useful for extraction of singular points. In general, the directional field at some location in the image is estimated by averaging the directions in a window around the desired location.

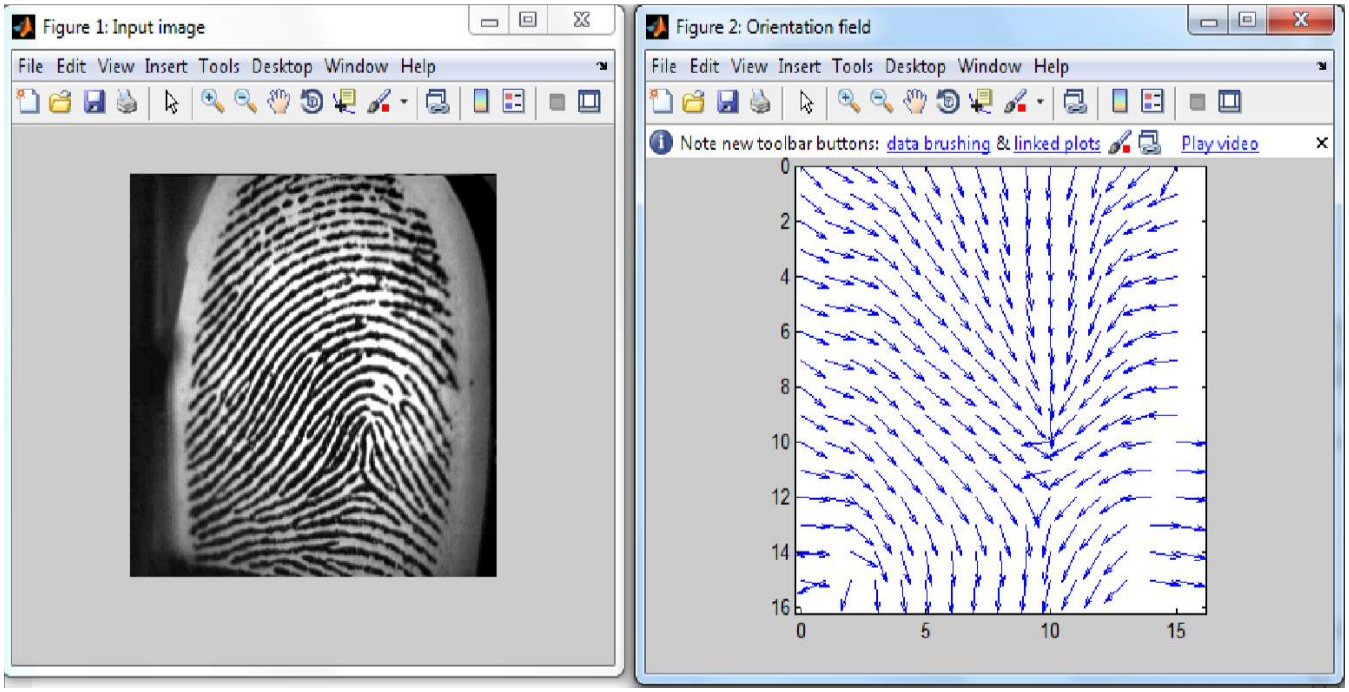


Figure 1: Input image
File Edit View Insert Tools Desktop Window Help
Figure 2: Orientation field
File Edit View Insert Tools Desktop Window Help
0
2
4
6
8
10
12
14
16
0
5
10
15

Note new toolbar buttons: data brushing & linked plots Play video

Figure 13: Orientation Field Estimation

Core point localization

Core points lie in the approximate centre of the finger impression. The core point is defined as the point where convex ridges have the maximum curvature. The core based match algorithm depends on core point to alignment the feature vector.

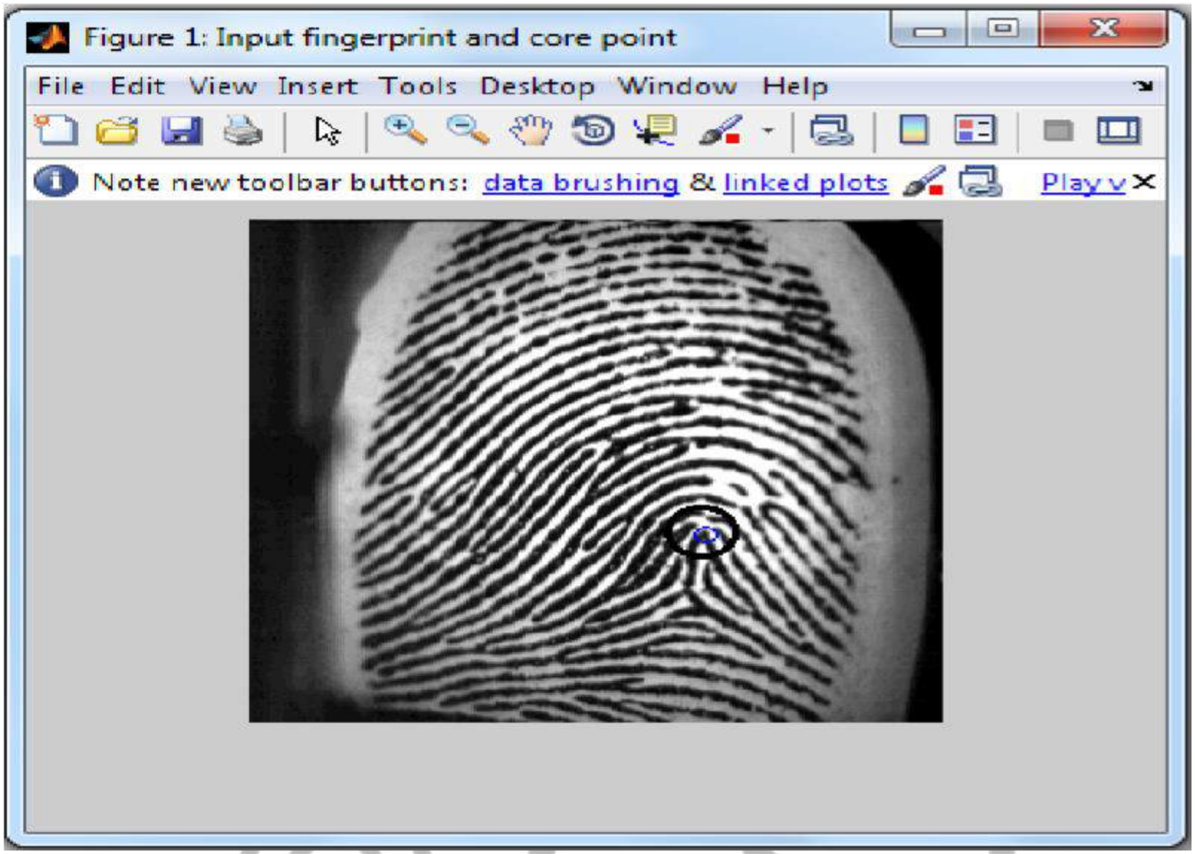


Figure 1: Input fingerprint and core point
File Edit View Insert Tools Desktop Window Help
Note new toolbar buttons: data brushing & linked plots
Play v

Figure 14: Core Point Localization

CONCLUSION

The fingerprints of different students were successfully enrolled and added to the database. The fingerprints were further verified and several dry runs were made that confirmed matches and mismatches for different samples. Apart from that, in this journal the FRT in MATLAB was used to demonstrate the various functions and processing methods used in image processing of the fingerprint. The outputs for all the trial runs and process demonstration were recorded.

The data transfer was made across a wireless channel in the lab connecting two terminals. Wireless communication meant that the range was limited to a short span but the data transfer process was efficient enough for the successful functioning of the system.

FUTURE WORK

There is a lot of scope in the field of biometrics application at the work place. The attendance system using fingerprint recognition can be of real use if certain nuances are taken into consideration. The wireless channel used was limited to a short range and hence the system could only be tested in the lab. For a greater range and more versatile application, a different channel could be considered which would ensure faster data transfer and provide better flexibility. The security aspect of transmission can be worked upon since data security in case of sensitive data transfer is highly essential.

Finally, the proposed model for each CHM and the PC server client software management system can be materialized using cost effective products offered in the market.

REFERENCES

[1] Zhang Yongqiang and Liu Ji, The design of wireless fingerprint attendance system, Proceedings of ICCT '06, International Conference on Communication Technology, 2006.

[2] Younhee Gil, Access Control System with high level security using fingerprints, IEEE the 32nd Applied Imagery Pattern Recognition Workshop (AIPR '03)

[3] Jain, A.K., Hong, L., and Bolle, R (1997), "On-Line Fingerprint Verification," IEEE Trans. On Pattern Anal and Machine Intell, 19(4), pp. 302-314.

[4] D.Maio and D. Maltoni. (1997), Direct gray-scale minutiae detection in fingerprints. IEEE Trans. Pattern Anal. And Machine Intell. 19(1):27-40, 1997. \

[5] Lee, C.J., and Wang, S.D. (1999), Fingerprint feature extraction using Gabor filters, Electron. Lett. 1999, 35, (4), pp.288-290.

[6] L. Hong, Y. Wan and A.K. Jain, (1998), "Fingerprint Image Enhancement: Algorithms and Performance Evaluation", IEEE Transactions on PAMI, Vol. 20, No. 8, pp.777-789, August 1998.

[7] SPRA894A, Texas Instruments, DSP for Smart Biometric Solutions

[8] User Manual, DWA-510

[9] SPRAA23, Texas Instruments, FADT2 Quick Start Guide

[10] TMS320C6713 DSK Technical Reference, (506735-0001 Rev. B)

[11] FVC2002. http://bias.csr.unibo.it/fvc2002/

[12] Fingerprint Recognition System by Luigi Rosa.

[13] Shlomo Greenberg, Mayer Aladjem, Daniel Kogan and Itshak Dimitrov, Fingerprint Image Enhancement using Filtering Techniques.

APPENDIX

List of some pseudo codes studied and developed for software implementation of various functions.

Pseudo code 1: Enhancement of Image

```
function [fImage]=im_enhance(InImage,f)

Im = 255-double(image);

[a,b] = size(Im);

// Apply floor function to round values of a & b; say to a1,b1

In = zeros(a1,b1); //
for 32 bit pixel data
for i=1:32:a1 for
j=1:32:b1

// calculate convolution based Fast Fourier
Transform Fim=fft2( Im(i:m,j:n) ); factor=abs(Fim).^f;

// find inverse DFT of F vector multiplied with factor

Imdata= abs(ifft2(Fim.*factor));

// Normalise the obtained Imdata by dividing each element with the max. Value

In(i:m,j:n) = normalized_Imdata;

// Obtain Histogram Equalisation of image

Fimage=In*255;

Fimage=histeq(Fimage); // improves contrast of image by transforming intensity image
```

Pseudo code 2: Binarization of Image

```
function [out] = im_bin_at(im,W);

//Image is segmented and adaptive threshold is calculated

// Initialize size matrix [a,b] & output matrix cut

// With step length W, divide it into blocks

//for loop for i -> 0 to a & j -> 0 to b, find mean threshold
m_thres = 0; if i+W-1 <= a & j+W-1 <= b m_thres =
mean2(im(i:i+W-1,j:j+W-1)); m_thres = 0.8*m_thres;

//calculate output matrix using m_thres as the threshold //scale
data to colomap defined in case of 2 input arguments
imagesc(out); colormap(gray);
```

Pseudo code 3: Estimation of block direction

```
function [d,z] = bl_dir(Im,blsize)

// image Im is obtained from the binarization function with defined blocksize

// initialize size [a,b]& direction matrix ‘direction’, gradient matrices W =
blsize; theta = 0; sum = 1; bg_present = 0;

bl_Index = zeros(ceil(a/W),ceil(b/W)); // find out
the filter gradient using sobel filter filter_grad =
fspecial('sobel'); // for  x-gradient

I_x = filter2(filter_grad,Im); %for y-
gradient
```

```
filter_grad = transpose(filter_grad);

I_y = filter2(filter_grad,Im);

(loop)

// update gradient matrices and obtain the sum, subtract and no. of times value if
sum ~= 0 & times ~=0

bg_present = (times *times + minus *minus)/(W*W*sum); if
bg_present > 0.05 blockIndex(ceil(i/W),ceil(j/W)) = 1;

// Obtain value for theta from inverse tan operation on subtract & times value as limits //
find center of the image by using rounded values in x & y dir and angle value ‘theta’ center
= [center;[round(i + (W-1)/2),round(j + (W-1)/2),theta]];

(end)

//scale the direction image & transform from polar to Cartesian coordinates along with
velocity vectors imagesc(direction);

[u,v] = pol2cart(center(:,3),n); // n= 16 or user defined quiver(center(:,2),center(:,1),u,v,);

// obtain z from morphological operations and b from perimeter pixels of z.
```