PALM PRINT RECOGNITION

B.Tech Project

Department of Electrical Engineering

Indian Institute of Technology, Ropar.

Supervisor
Dr. Subrahmanyam Murala

Submitted by:

Chennur Meghashyam (2011EE1050)
Suprith Balasubramanian(2011EE1073)
Vishal GJ (2011EE1078)

Certificate

We hereby declare that the results embodied in this dissertation entitled Palm Print Recognition is carried out by us during the year 2014-2015 in partial fulfillment of the award of B.Tech(Electrical Engineering) from Indian Institute of Technology , Ropar. We have not submitted the same to any other university or organization for the award of any other degree.

Meghashyam Chennur (2011EE1050)
Suprith Balasubramanian (2011EE1073)
Vishal GJ (2011EE1078)

Acknowledgement

I take immense pleasure by thanking Dr. Subrahmanyam Murula , Assistant Professor , Dept. of Electrical Engineering , Indian Institute of Technology , Ropar for guiding us throughout our B.Tech Project (BTP) work.

I wish to express my deep sense of gratitude to Dr . J S . Sahambi (HOD Electrical Engineering) and Dr. Ravi Babu Mulaveesala (Assistant Professor) for asking us valuable questions during our BTP presentation which gave us deep insight into our project.

I would also like to thank Dr. Rohit . Y .Sharma (Assistant Professor) for keeping us updated with the BTP presentation schedule and report submissions.

Finally , yet importantly I would like to express my heartfelt thanks to my beloved friends and batch mates for their valuable inputs and guidance to clear our doubts and for successful completion of our project

Meghashaym Chennur
Suprith Balasubramanian
Vishal GJ

TABLE OF CONTENTS

Abstract:

In this B.Tech Project a new algorithm for 'Palm Print Recognition' is proposed based on local feature extraction techniques which form the basis for an efficient palm print recognition method. The standard local binary pattern utilizes the relationship between the referenced pixel and its surrounding neighbors, whereas the proposed method encodes the relationship among the surrounding neighbors for a given referenced pixel in an image and also takes into account the relationship between the referenced pixel and its neighbors. The Local Mesh Pattern (LMeP) technique which is used, takes the difference between the adjacent and the alternate neighboring pixels for the extraction technique. Here, we have proposed a new feature descriptor for palm print recognition. Results will be compared with the existing state-of-the-art features for palm print recognition in terms of their evaluation measures.

Introduction

Biometric technology is used for automatic personal recognition based on biological traits—fingerprint, iris, face, palm print, hand geometry, vascular pattern, voice—or behavioral characteristics—gait, signature, typing pattern. Fingerprinting is the oldest of these methods and has been utilized for over a century by law enforcement officials who use these distinctive characteristics to keep track of criminals[1].

The National Science and Technology Council provides the following overview of biometric system components: "A typical biometric system is comprised of five integrated components: A sensor is used to collect the data and convert the information to a digital format. Signal processing algorithms perform quality control activities and develop the biometric template. A data storage component keeps information that new biometric templates will be compared to. A matching algorithm compares the new biometric template to one or more templates kept in data storage. Finally, a decision process (either automated or human-assisted) uses the results from the matching component to make a system-level decision."[2]

Authentication systems can be based on three measures: what you know-a password, what you have- a token or pass card, or what you are-biometrics. Passwords, keys and tokens can be forgotten, lost, stolen or otherwise compromised. Biometric identifiers also carry risks. Engineering professor, Tsutomu Matsumoto, demonstrated this point by using a digital camera, a PC, and gelatin to fashion a fake finger which fooled biometric scanners 80% of the time. [3] However, new applications can detect fakes by identifying sweat pores, measuring conduction properties, and determining the differences in how a live finger and a dummy finger deform the surface of a sensor.[4]

Biometric systems are vulnerable to two types of failures: a false-positive, in which a system falsely identifies an imposter as the valid user, and a false-negative, in which the system fails to make a match between a valid user and the stored template. Because no single identifier is fool-proof, using more than one method, such as a biometric measure in addition to a personal identification number, can enhance security.

Government uses of biometrics include the US-VISIT program, in which visa-issuing consular offices collect biometric data, finger scans and photographs, which are checked

---

[1] "Biometric Personal Authentication Using Keystroke Dynamics: A Review."
[2] "Biometrics.pdf."
[3] Ibid.
[4] "BBC News | SCI/TECH | Doubt Cast on Fingerprint Security."

against a database of known criminals and suspected terrorists. The traveler’s identity is verified at entry and exit of the country. Integrated Automated Fingerprint Identification System (IAFIS), the FBI’s national fingerprint and criminal history system, the Transportation Workers Identification Credentials (TWIC) program, and the Registered Traveler (RT) program are other government uses of biometrics.

The following information from the National Science and Technology Council presents four types of biometric standards: technical interfaces, data interchange formats, application profile standards, and performance testing and reporting.

Technical interface standards specify interfaces and interactions between biometric components and sub-systems, including the possible use of security mechanisms to protect stored data and data transferred between systems; and specify the architecture and operation of biometric systems in order to identify the standards that are needed to support multi-vendor systems and their applications. Data Interchange Formats specify the content, meaning, and representation of formats for the interchange of biometric data, e.g., Finger Pattern Based Interchange Format, Finger Minutiae Format for Data Interchange, Face Recognition Format for Data Interchange, Iris Interchange Format, Finger Image Based Interchange Format, Signature/Sign Image Based Interchange Format, and Hand Geometry Interchange Format; and specify notation and transfer formats that provide platform independence and separation of transfer syntax from content definition. Application Profile Standards specify one or more base standards and standardized profiles, and where applicable, the identification of chosen classes, conforming subsets, options, and parameters of those base standards or standardized profiles necessary to accomplish a particular function. “Performance Testing and Reporting standards specify biometric performance metric definitions and calculations, approaches to test performance, and requirements for reporting the results of these tests.

There are several concerns surrounding the use of biometrics for identification. If a credit card or key is lost or stolen, the card can be cancelled; the locks can be changed and replaced. However, if biometric data is compromised, there are a finite number of replacements, as a person has only 10 fingers, two eyes, etc. Another concern is the possibility that sensors which require contact could be unsanitary. Ensuring the privacy and security of biometric data is also of concern, as users will be unlikely to accept the technology if information could potentially be tampered with, stolen or otherwise misused. [5]

[5] “A Survey of Palmprint Recognition.”

The inner surface of the palm normally contains three flexion creases, secondary creases and ridges. The flexion creases are also called principal lines and the secondary creases are called wrinkles. Although the three major flexions are genetically dependent, most of other creases are not. Even identical twins have different palm prints. These non-genetically deterministic and complex patterns are very useful in personal identification. Scientists know that palm lines are associated with some genetic diseases including Down’s syndrome, Aarskog syndrome, Cohen syndrome and fetal alcohol syndrome.

History

In many instances throughout history, examination of handprints was the only method of distinguishing one illiterate person from another since they could not write their own names. Accordingly, the hand impressions of those who could not record a name but could press an inked hand onto the back of a contract became an acceptable form of identification. In 1858, Sir William Herschel, working for the Civil Service of India, recorded a handprint on the back of a contract for each worker to distinguish employees from others who might claim to be employees when payday arrived. This was the first recorded systematic capture of hand and finger images that were uniformly taken for identification purposes[6].

The first known AFIS system built to support palm prints is believed to have been built by a Hungarian company. In late 1994, latent experts from the United States benchmarked the palm system and invited the Hungarian company to the 1995 International Association for Identification (IAI) conference. The palm and fingerprint identification technology embedded in the palm system was subsequently bought by a US company in 1997[7].

In 2004, Connecticut, Rhode Island and California established statewide palm print databases that allowed law enforcement agencies in each state to submit unidentified latent palm prints to be searched against each other's database of known offenders. Australia currently houses the largest repository of palm prints in the world. The new Australian National Automated Fingerprint Identification System (NAFIS) includes 4.8 million palm prints[8].

[6] “Palm Print Recognition.pdf.”
[7] “Palm Print - Wikipedia, the Free Encyclopedia.”
[8] “Palmprint Verification Using Binary Orientation Co-Occurrence Vector.”

The new NAFIS complies with the ANSI/NIST international standard for fingerprint data exchange, making it easy for Australian police services to provide fingerprint records to overseas police forces such as Interpol or the FBI, when necessary. Over the past several years, most commercial companies that provide fingerprint capabilities have added the capability for storing and searching palm print records. While several state and local agencies within the US have implemented palm systems, a centralized national palm system has yet to be developed.

Currently, the Federal Bureau of Investigation (FBI) Criminal Justice Information Services (CJIS) Division houses the largest collection of criminal history information in the world. This information primarily utilizes fingerprints as the biometric allowing identification services to federal, state, and local users through the Integrated Automated Fingerprint Identification System (IAFIS). The Federal Government has allowed maturation time for the standards relating to palm data and live-scan capture equipment prior to adding this capability to the current services offered by the CJIS Division.

The FBI Laboratory Division has evaluated several different commercial palm AFIS systems to gain a better understanding of the capabilities of various vendors. Additionally, state and local law enforcement have deployed systems to compare latent palm prints against their own palm print databases. It is a goal to leverage those experiences and apply them towards the development of a National Palm Print Search System.

In April 2002, a Staff Paper on palm print technology and IAFIS palm print capabilities was submitted to the Identification Services (IS) Subcommittee, CJIS Advisory Policy Board (APB). The Joint Working Group then moved “for strong endorsement of the planning, costing, and development of an integrated latent print capability for palms at the CJIS Division of the FBI. This should proceed as an effort along the same parallel lines that IAFIS was developed and integrate this into the CJIS technical capabilities.

As a result of this endorsement and other changing business needs for law enforcement, the FBI announced the Next Generation IAFIS (NGI) initiative. A major component of the NGI initiative is the development of the requirements for and deployment of an integrated National Palm Print Service. Law enforcement agencies indicate that at least 30 percent of the prints lifted from crime scenes — from knife hilts, gun grips, steering wheels, and window panes — are of palms, not fingers.6 For this reason, capturing and scanning latent palm prints is becoming an area of increasing interest among the law enforcement community.

The National Palm Print Service is being developed on the basis of improving law enforcement’s ability to exchange a more complete set of biometric information, making additional identifications, quickly aiding in solving crimes that formerly may have not been possible, and improving the overall accuracy of identification through the IAFIS criminal history records.

Theory

i. Biometric System

Palm print research employs either high or low resolution images. High resolution images are suitable for forensic applications such as criminal detection. Low resolution images are more suitable for civil and commercial applications such as access control. Generally speaking, high resolution refers to 400 dpi or more and low resolution refers to 150 dpi or less. Researchers can extract ridges, singular points and minutia points as features from high resolution images while in low resolution images they generally extract principal lines, wrinkles and texture. Initially palm print research focused on high-resolution images but now almost all research is on low resolution images for civil and commercial applications[9].

The design of a biometric system takes account of five objectives: cost, user acceptance and environment constraints, accuracy, computation speed and security (Fig. 1). Reducing accuracy can increase speed. Typical examples are hierarchical approaches. Reducing user acceptance can improve accuracy. For instance, users are required to provide more samples for training. Increasing cost can enhance security[10]. We can embed more sensors to collect different signals for liveness detection. In some applications, environmental constraints such as memory usage, power consumption, size of templates and size of devices have to be fulfilled. A biometric system installed in PDA (personal digital assistant) requires low power and memory consumption but these requirements may not be vital for biometric access control systems. A practical biometric system should balance all these aspects.

"Palm Print - Wikipedia, the Free Encyclopedia."

[9] Ibid.

[10] "IEEE Xplore Abstract - Dominant Local Binary Patterns for Texture Classification."

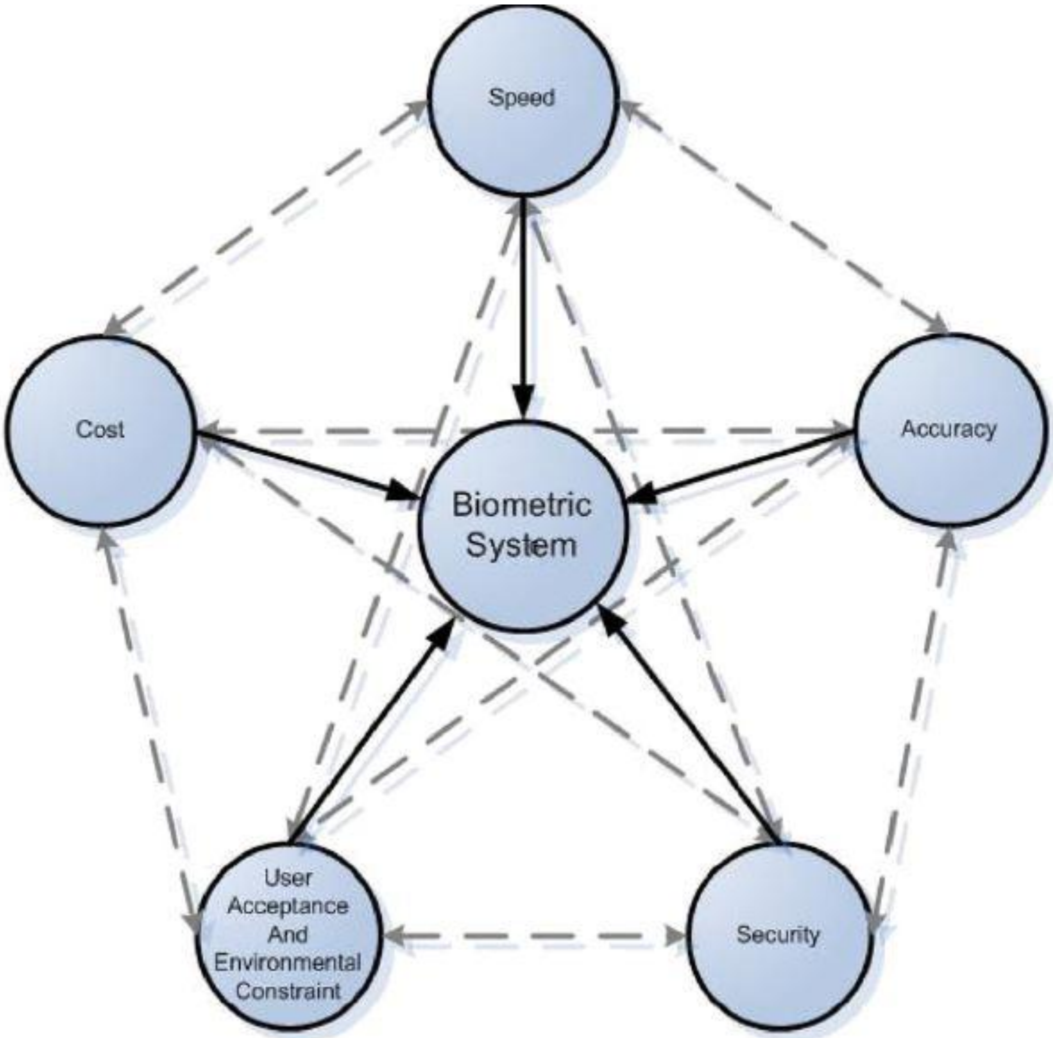


Fig 1: Inter relationships between different objectives for designing a biometric system[11]

A typical palm print recognition system consists of five parts: palm print scanner, preprocessing, feature extraction, matcher and database illustrated in Fig. 2. The palm print scanner collects palm- print images. Preprocessing sets up a coordinate system to align palm print images and to segment a part of palm print image for feature extraction. Feature extraction obtains effective features from the preprocessed palm prints. A matcher compares two palm print features and a database stores registered templates.

[11] "A Survey of Palmprint Recognition."

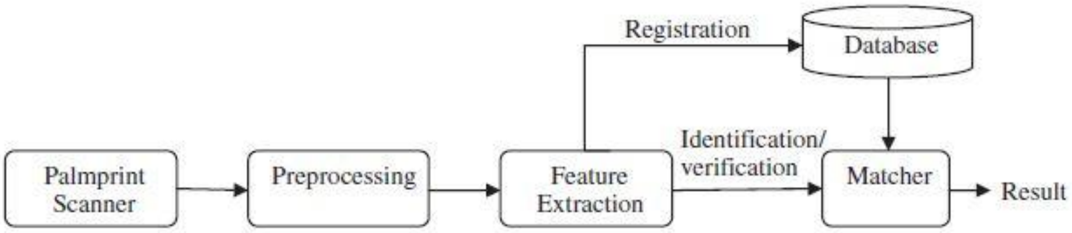


Fig 2: An illustration of a typical palm print recognition system[12]

ii. Palm Print Scanners

For research purpose generally four types of sensors are utilized: CCD-based palm print scanners, digital cameras, digital scanners and video cameras to collect palm print images. CCD-based palm print scanners capture high quality palm print images and align palms accurately because the scanners have pegs for guiding the placement of hands. These scanners simplify the development of recognition algorithms because the images are captured in a controlled environment.

However, developing a CCD-based palm print scanner requires a suitable selection of lens, camera, and light sources. Although these palm print scanners can capture high quality images, they are large. Collection approaches based on digital scanners, digital cameras and video cameras require less effort for system design and can be found in office environments.

These approaches do not use pegs for the placement of hands. Some researchers believe that this increases user acceptance. Digital and video cameras can be used to collect palm print images without contact, an advantage if hygiene is a concern. However, these images might cause recognition problem as their quality is low because they collect is in an uncontrolled environment with illumination variations and distortions due to hand movement. Digital scanners are not suitable for real-time applications because of the scanning time[13].

iii. Preprocessing

Preprocessing is used to align different palm print images and to segment the center for feature extraction. Most of the preprocessing algorithms employ the key points between fingers to set up a co- ordinate system. Preprocessing involves five common steps: (1)

[12] Ibid.
[13] Ibid.

binarizing the palm images, (2) extracting the contour of hand and/or fingers, (3) detecting the key points, (4) establishing a coordination system and (5) extracting the central parts.

The first and second steps in all the preprocessing algorithms are similar. However, the third step has several different implementations including tangent, bisector and finger-based to detect the key points between fingers. The tangent-based approach considers the two boundaries—one from point finger and middle finger and the other from ring finger and last finger—as two convex curves and computes the tangent of these two curves. The two intersections are considered as two key points for establishing the coordinate system.

Tangent-based approaches have several advantages. They depend on a very short boundary around the bottom of fingers. Therefore, it is robust to incomplete fingers (as in the disabled) and the presence of rings. Bisector-based approach constructs a line using two points, the center of gravity of a finger boundary and the midpoint of its start and end points. The intersection of the line and the finger boundary is considered a key point. Han and his team propose two approaches to establish the coordinate system, one based on the middle finger and the other based on the point, middle and ring fingers.

The middle finger approach uses a wavelet to detect the fingertip and the middle point in the finger bottom and construct a line passing through these two points . The multiple finger approach uses a wavelet and a set of predefined boundary points on the three fingers to construct three lines in the middle of the three fingers. The two lines from point and ring fingers are used to set the orientation of the coordinate system and the line from the middle finger is used to set its position. These approaches use only the information on the boundaries of fingers while Kumar et al. proposed using all information in palms. They fit an ellipse to a binary palm print image and set up the coordinate system according to the orientation of the ellipse[14].

Palm print recognition techniques

Once the central part is segmented, features can be extracted for matching. There are two types of recognition algorithms, verification and identification. Verification algorithms must be accurate. Identification algorithms must be accurate and fast (matching speed). Verification algorithms are line- , subspace- and statistic-based. Some algorithms here can support a certain scale of identification.

There are many feature extraction techniques proposed for palm print recognition such as LBP, LDP, LTP, LMeP. Our research interest is LBP and LMeP.

[14] Ibid.

METHODOLOGY

1. Local Binary Patterns (LBPs)

The LBP feature vector is calculated for each pixel in a cell by comparing it with each of its 8 neighbors. The original LBP operator [5] forms labels for the image pixels by thresholding the 3 x 3 neighborhood of each pixel with the center value and considering the result as a binary number.

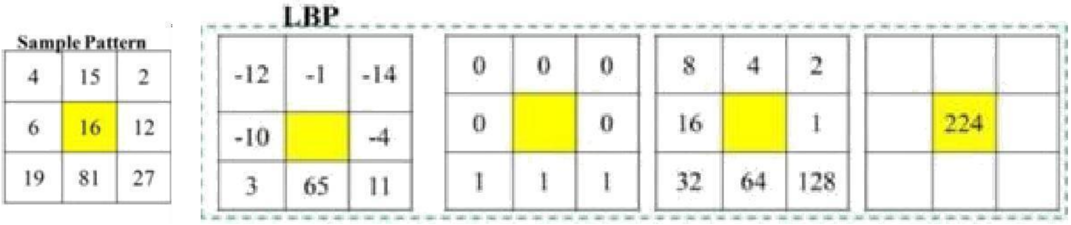


Figure 3: LBP operation[15]

Mathematically, ic and ip (p=0 to P-1) denote the gray value of the center pixel and gray value of the neighbor pixel on a circle of radius R, respectively, and P is the number of the neighbors.

The value of the LBP code of a pixel (x_c, y_c) is given by:

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^p \qquad s(x) = \begin{cases} 1, if\ x \ \geq \ 0; \\ 0, otherwise. \end{cases}$$

Approach

The method that we proposed combines the feature maps from the Local Mesh patterns as well as the Local Binary pattern feature maps. In the first step, edge information was obtained by calculating the difference in grayscale values among the neighbors for a given center pixel in the image. The equation below gives an example of calculating LMeP values.

[15] "IEEE Xplore Abstract - Local Mesh Patterns Versus Local Binary Patterns: Biomedical Image Indexing and Retrieval."

$$
\begin{aligned}
\mathrm{LMeP}^{j}_{P,R} &= \sum_{i=1}^{P} 2^{(i-1)} \times f_1(g_\alpha|_R - g_i|_R) \\
\alpha &= 1 + \mathrm{mod}((i + P + j - 1), P) \\
\forall j &= 1, 2, .., (P/2)
\end{aligned}
$$

Where *j* represents the LMeP [5] index and mod (*x*, *y*) returns the reminder for *x*/*y* operation. Here, we use a circular binary representation referred to as uniform pattern to reduce the computational cost. After identifying the local pattern, the whole image is represented by building a histogram using

$$
H_S(l) = \frac{1}{N_1 \times N_2} \sum_{j=1}^{N_1} \sum_{k=1}^{N_2} f_2(\mathrm{PTN}(j,k), l); \quad l \in [0, P(P-1)+2]
$$

$$
f_2(x,y) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{else} \end{cases}
$$

Where N1xN2 represents the size of an input image.

The whole image is represented by histograms. The histograms obtained through the LBP and LMeP can be combined to get better edge information from the image.

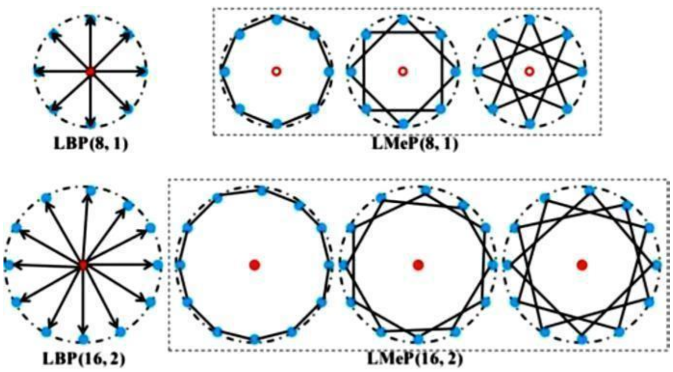


Figure 4. LBP and the first three LMeP calculations for a given (*P*, *R*).[16]

[16] Ibid.

Experiments and Discussions

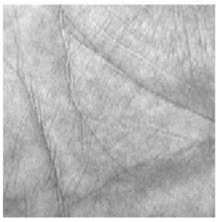
After Regions of Interest (ROIs) are obtained from the pre-processing step, important features were extracted using the method discussed above. The database used for query matching contains 7640 images, with 20 samples per person.

The **algorithm** used for the proposed method is as follows:

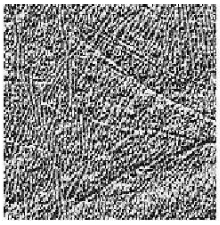
1. Load the gray scale image (if image is in RGB, convert it to grayscale).
2. For each center pixel, calculate the first three LMeP operators.
3. Calculate the local differences among the neighbor pixels.
4. Calculate the local differences between the center pixel and each of the neighbor pixels.
5. Calculate the binary patterns.
6. Construct the histograms based on the uniform binary patterns.
7. Form a feature vector by concatenating four histograms.

Now that we have obtained the feature vector of the image matching it would give fairly good results. But in order to get better accuracy we constructed localized histograms.

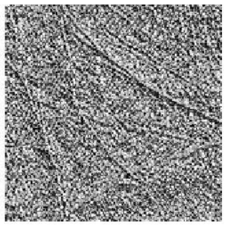
LBP and LMeP for a sample image



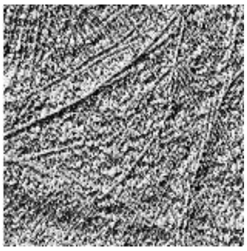
Sample image



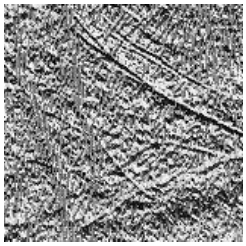
LBP image



LMeP (1^{st} neighbor)



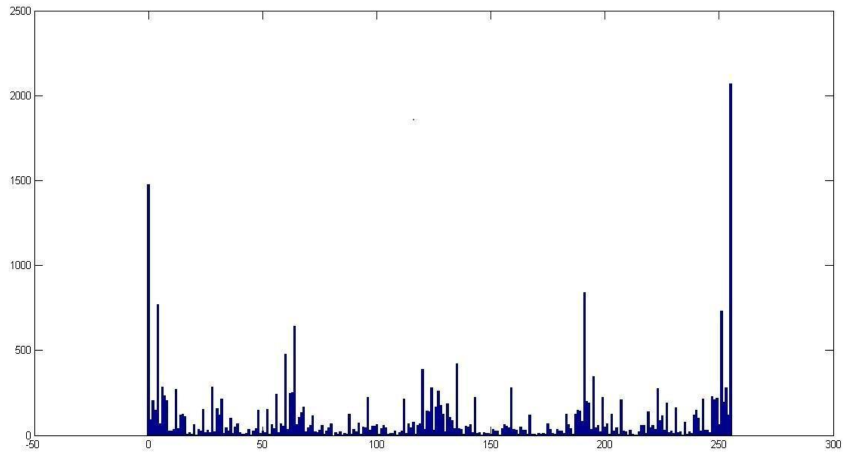
LMeP (2^{nd} neighbor)



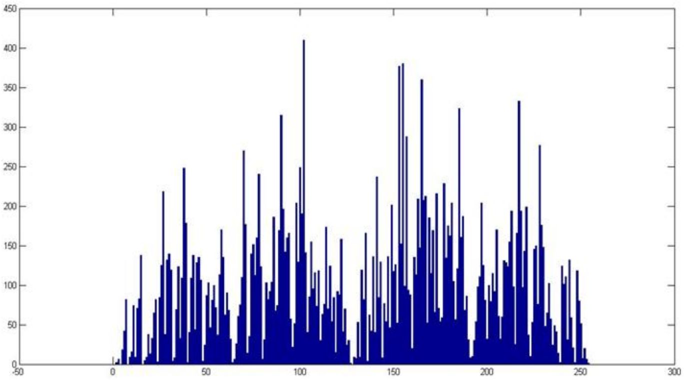
LMeP (3^{rd} neighbor)

Histograms:

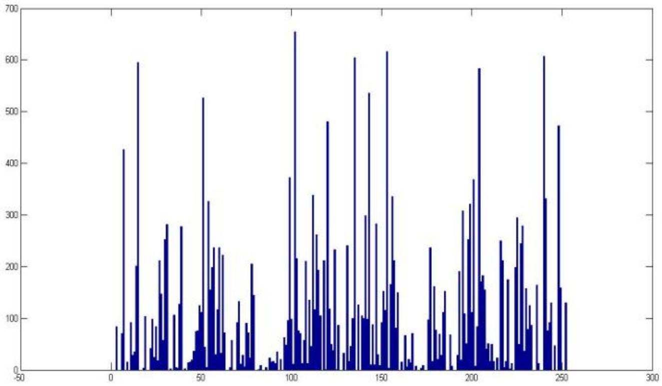
LBP (X axis: number of pixels Y axis: pixel intensity)



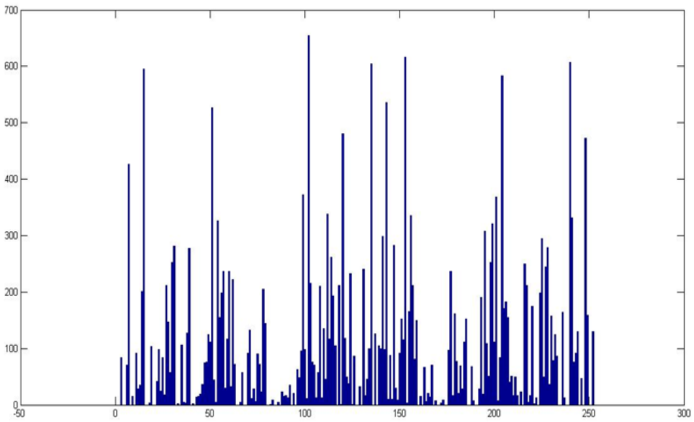
LMeP- I



LMeP -
II



LMeP -
III



Localized Algorithm

The technique of localized histograms was used to improve the accuracy. The image was divided into small blocks and histograms were calculated for each of the smaller blocks. The final feature vector was calculated by concatenating these histograms. Although this method resulted in higher accuracy of matching data, the boundary information of the smaller histograms was being lost. In order to overcome this problem, the newly formed smaller images were overlapped.

Matching

The feature vector for query image Q represented as $fQ = (fQ_1, fQ_2. \ . \ . \ fQ_{L\,g})$, is obtained from feature extraction. Similarly each image in the database is represented with feature vector
$f\mathrm{DB}_j = (f\mathrm{DB}_{j\,1}, f\mathrm{DB}_{j\,2} \ldots f\mathrm{DB}_{j\,L\,g})$; $j = 1, 2. \ . \ . \ |\mathrm{DB}|$. The goal is to select the n best images that resemble the query image. This involves selection of n top matched images by measuring the distance between query image and images in the database |DB|.

$$D(Q_i, \mathrm{DB}_{ji}) = \sum_{i=1}^{Lg} \left| \frac{f_{\mathrm{DB}_{ji}} - f_{Q_i}}{1 + f_{\mathrm{DB}_{ji}} + f_{Q_i}} \right|$$

where fDBji is ith feature of jth image in the database |DB|.

The database includes palm prints of both the left and right palms of 100 persons, 10 images for each palm while the query image database includes 2 palm prints of each person (one left and one right). The remaining images form the database for testing, with 9 images of each palm for a person After the distance metric is calculated for each query image, the 'n' best images resembling query image are selected corresponding to the 'n' least values of the distance metric.

For each query, the system collects n database images $X = (x1, x2. \ . \ . \ xn)$, with the shortest image matching distance is given by (16). If xi ; $i = 1, 2. \ . \ . \ n$ belong to the same category of the query image, we say the system has correctly matched the desired[17].

---

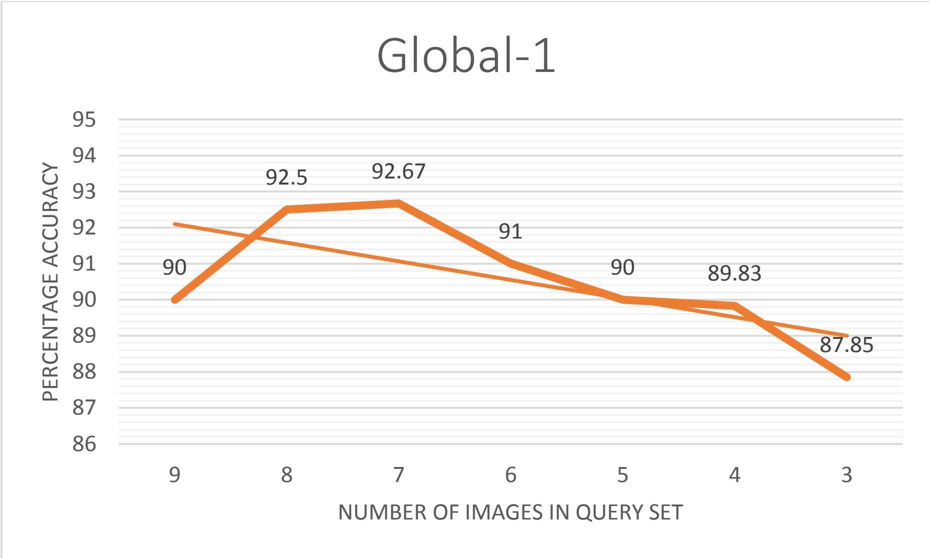
[17] Ibid.

Results and Analysis:

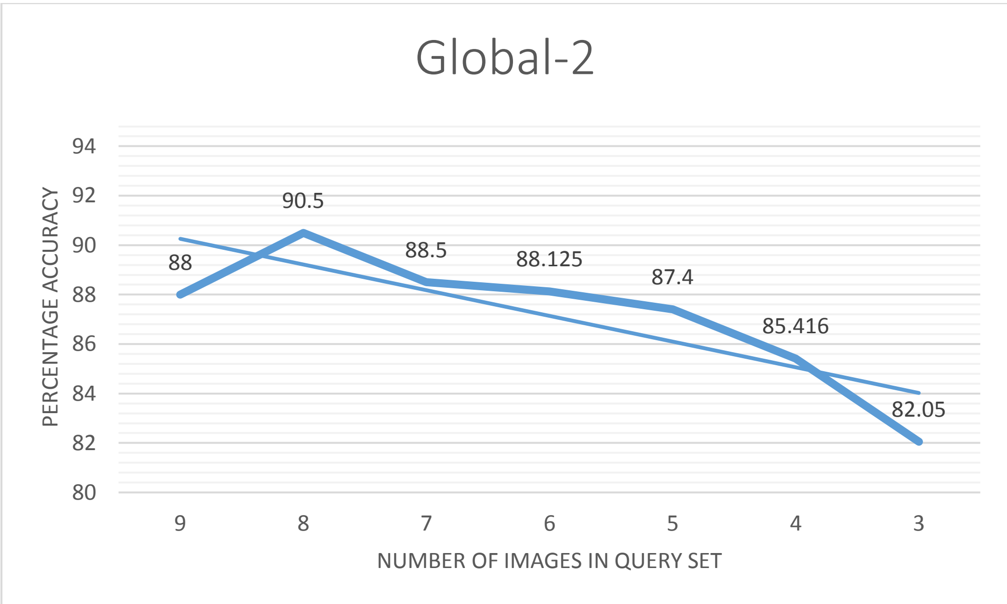
There were 2 techniques used to calculate the concatenated histograms:
Global method, where the feature vectors were calculated for the entire image without any overlapping.
Local method, where feature vector calculation was done for individual blocks of the image with overlapping of pixels.

The average accuracy was calculated by varying the number of images in the query image set with the number of top matched images being one and two.

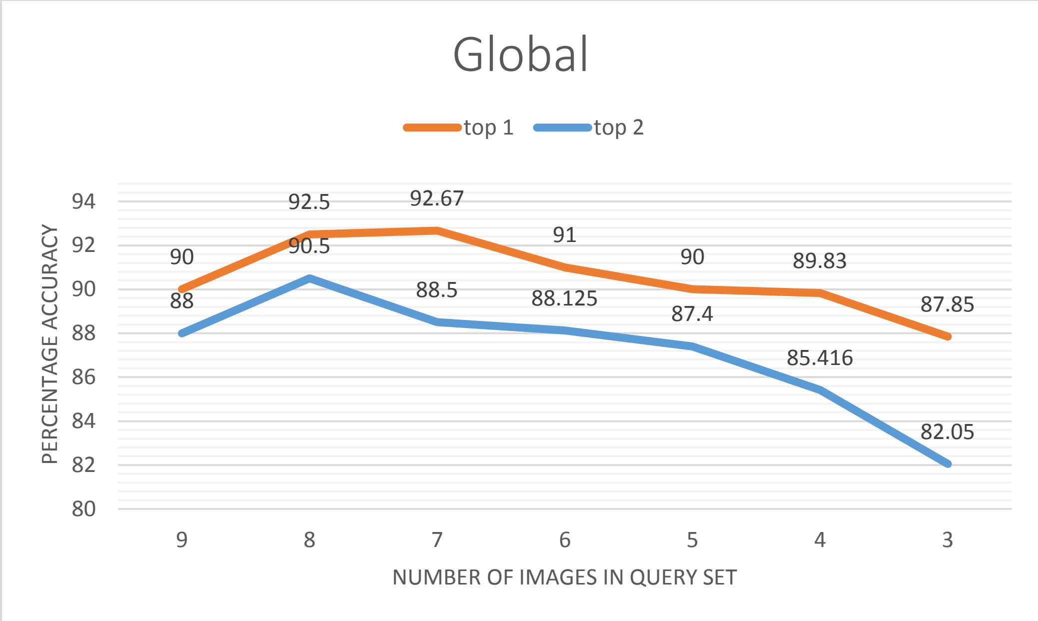
Avg. Accuracy				
No. of persons	Matching type	Accuracy	No. of persons in check set	No. considered
100	global	88	9	2
100	global	90.5	8	2
100	global	88.5	7	2
100	global	88.125	6	2
100	global	87.4	5	2
100	global	85.416	4	2
100	global	82.05	3	2
100	global	90	9	1
100	global	92.5	8	1
100	global	92.67	7	1
100	global	91	6	1
100	global	90	5	1
100	global	89.83	4	1
100	global	87.85	3	1



This graph shows average accuracy as a function of number of images in the query set for 1 top matched image



Avg. accuracy vs No. of images in query set for 2 top matched images

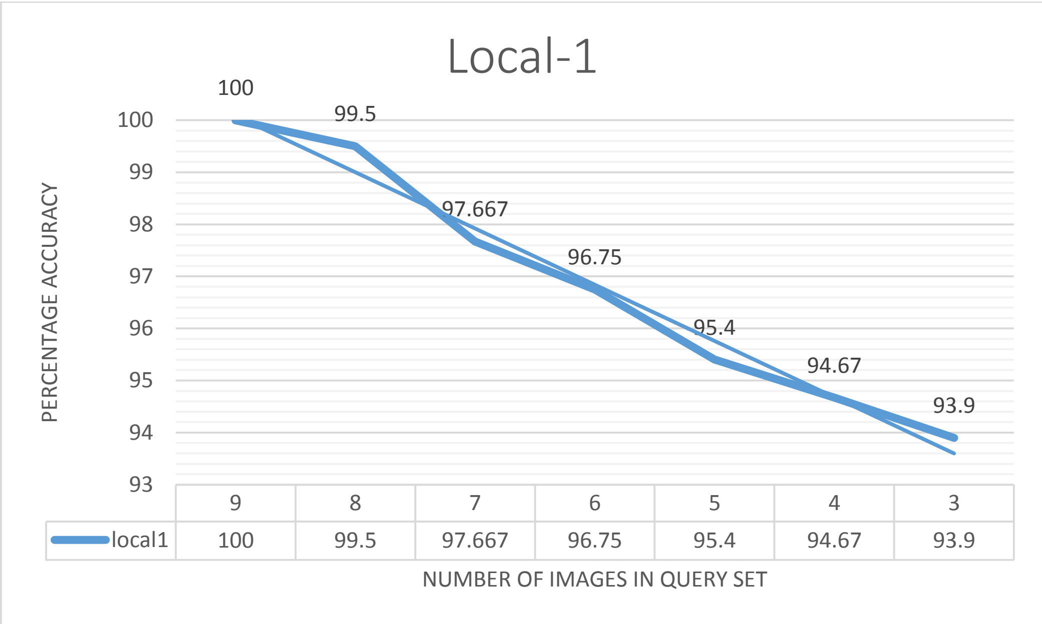


Comparison between Global-1 and Global-2

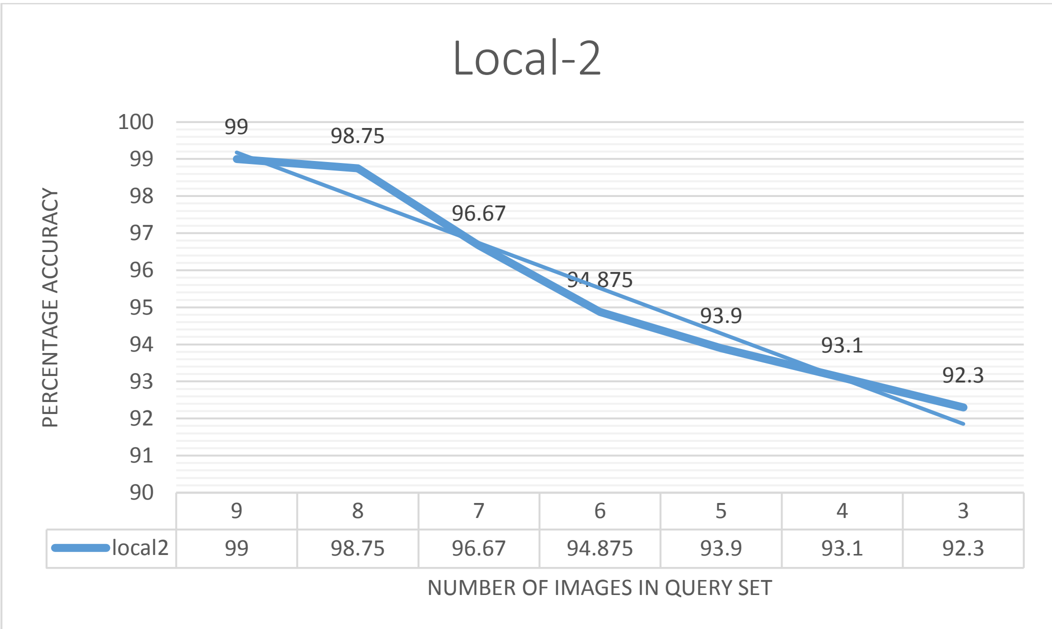
The average accuracy for 1 top matched image is higher than for 2 top matched images. From the graph, the average accuracy for the query set containing seven images is higher than for the rest of the cases.

Average accuracy for Local Method

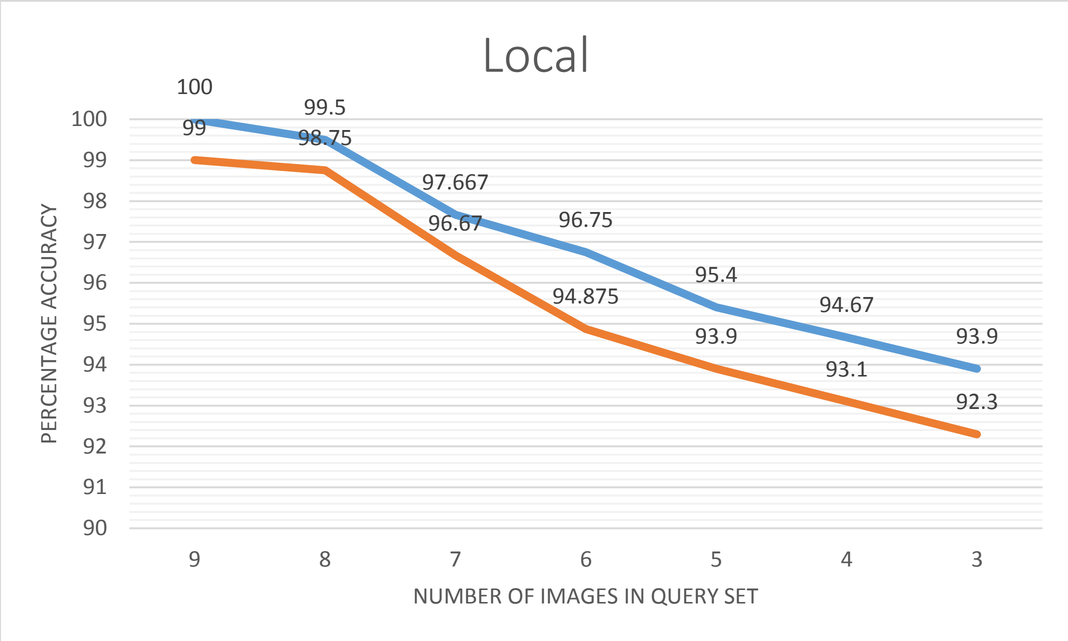
No. of persons	Matching type	Accuracy	No. of persons in check set	No. considered	Local Matrix size	overlap
100	local	99	9	2	16	2
100	local	98.75	8	2	16	2
100	local	96.67	7	2	16	2
100	local	94.875	6	2	16	2
100	local	93.9	5	2	16	2
100	local	93.1	4	2	16	2
100	local	92.3	3	2	16	2
100	local	100	9	1	16	2
100	local	99.5	8	1	16	2
100	local	97.667	7	1	16	2
100	local	96.75	6	1	16	2
100	local	95.4	5	1	16	2
100	local	94.67	4	1	16	2
100	local	93.9	3	1	16	2



Avg. accuracy vs No. of images in query set for 1 top matched image



Avg. accuracy vs No. of images in query set for 2 top matched images



Comparison between Local-1 and Local-2

The average accuracy for 1 top matched image is higher than for 2 top matched images, with the highest accuracy of 100% obtained for the query set containing 9 images.

A comparison of all the four methods used above

CONCLUSION

In this paper, the method we proposed which used a combination of both the Local Binary Pattern (LBP) and Local Mesh Pattern (LMeP) was tested on a database of 100 persons, 10 images for each person.
The technique of localized histograms was used in the implementation of the matching technique. The Local method for matching yielded maximum accuracy of 100% and the minimum accuracy obtained was 92.3%.

Appendix

Code for calculating feature vector database

Invoke Function

```
%%
clc
clear all
%% Forming a feature vector database

n = input('Please specify the total number of persons :  ');

a=1;
feature  = cell(n,1);

number=input('Please enter the checking set size :');
number1=(10-number)+1;

b=input('Choose a method to compute histogram database \n 1.Global \n 2.Local
:');
number1=(10-number)+1;

if (b==2)
    ls=input('Please enter the size of the local matrix = ');
    d=input('Please enter the number of overlap column = ');
end
fprintf('\n\nPlease wait . . . Creating Feature vector matrix.\n');
tic;


for i = 1:n

    fprintf('PERSON NO:=%d . . . Done.\n',i);
  for j= number1:10

    filename=strcat('C_PolyU_',int2str(i),'_F_',int2str(j),'.bmp');
    I=imread(filename);

    if (b==1)
        [ histogram ] = lbp_lmep( I );

    elseif (b==2)
        [ histogram ] = lbp_lmep_local( I , ls , d);
    end
    feature{a,1} = histogram;
    a=a+1;
  end
```

```
end


if (b==1)
    save('feature_database_global_F.mat','feature');
elseif(b==2)
    save('feature_database_local_F.mat','feature')
end

toc;
fprintf('The Feature vector database for %d persons has been created',n);
```

LBP_LMeP function

```
 function [ histogram ] = lbp_lmep( im )

[sa,sb]=size(im);
img=int32(im);

%% Forming Matrices for each matrix element
img1=img(2:sa-1,3:sb);
img2=img(1:sa-2,3:sb);
img3=img(1:sa-2,2:sb-1);
img4=img(1:sa-2,1:sb-2);
img5=img(2:sa-1,1:sb-2);
img6=img(3:sa,1:sb-2);
img7=img(3:sa,2:sb-1);
img8=img(3:sa,3:sb);
imgc=img(2:sa-1,2:sb-1);

%% Calculation for LBP

f1=img1-imgc;
f2=img2-imgc;
f3=img3-imgc;
f4=img4-imgc;
f5=img5-imgc;
f6=img6-imgc;
f7=img7-imgc;
f8=img8-imgc;

%% Calculation for LMep Index-1

f1_1=img2-img1;
f2_1=img3-img2;
f3_1=img4-img3;
f4_1=img5-img4;
```

```
f5_1=img6-img5;
f6_1=img7-img6;
f7_1=img8-img7;
f8_1=img1-img8;

%% Calculation for LMep Index-2

f1_2=img3-img1;
f2_2=img4-img2;
f3_2=img5-img3;
f4_2=img6-img4;
f5_2=img7-img5;
f6_2=img8-img6;
f7_2=img1-img7;
f8_2=img2-img8;

%% Calculation for LMep Index-3

f1_3=img4-img1;
f2_3=img5-img2;
f3_3=img6-img3;
f4_3=img7-img4;
f5_3=img8-img5;
f6_3=img1-img6;
f7_3=img2-img7;
f8_3=img3-img8;


%% Applying binary condition and Assigning weight for LBP

[ w1 ] = neighbour( f1,sa,sb,0 );
[ w2 ] = neighbour( f2,sa,sb,1 );
[ w3 ] = neighbour( f3,sa,sb,2 );
[ w4 ] = neighbour( f4,sa,sb,3 );
[ w5 ] = neighbour( f5,sa,sb,4 );
[ w6 ] = neighbour( f6,sa,sb,5 );
[ w7 ] = neighbour( f7,sa,sb,6 );
[ w8 ] = neighbour( f8,sa,sb,7 );

%% Applying binary condition and Assigning weight for LMep-1

[ w1_1 ] = neighbour( f1_1,sa,sb,0 );
[ w2_1 ] = neighbour( f2_1,sa,sb,1 );
[ w3_1 ] = neighbour( f3_1,sa,sb,2 );
[ w4_1 ] = neighbour( f4_1,sa,sb,3 );
[ w5_1 ] = neighbour( f5_1,sa,sb,4 );
[ w6_1 ] = neighbour( f6_1,sa,sb,5 );
[ w7_1 ] = neighbour( f7_1,sa,sb,6 );
[ w8_1 ] = neighbour( f8_1,sa,sb,7 );
```

```
%% Applying binary condition and Assigning weight for LMep-2

[ w1_2 ] = neighbour( f1_2,sa,sb,0 );
[ w2_2 ] = neighbour( f2_2,sa,sb,1 );
[ w3_2 ] = neighbour( f3_2,sa,sb,2 );
[ w4_2 ] = neighbour( f4_2,sa,sb,3 );
[ w5_2 ] = neighbour( f5_2,sa,sb,4 );
[ w6_2 ] = neighbour( f6_2,sa,sb,5 );
[ w7_2 ] = neighbour( f7_2,sa,sb,6 );
[ w8_2 ] = neighbour( f8_2,sa,sb,7 );

%% Applying binary condition and Assigning weight for LMep-3

[ w1_3 ] = neighbour( f1_3,sa,sb,0 );
[ w2_3 ] = neighbour( f2_3,sa,sb,1 );
[ w3_3 ] = neighbour( f3_3,sa,sb,2 );
[ w4_3 ] = neighbour( f4_3,sa,sb,3 );
[ w5_3 ] = neighbour( f5_3,sa,sb,4 );
[ w6_3 ] = neighbour( f6_3,sa,sb,5 );
[ w7_3 ] = neighbour( f7_3,sa,sb,6 );
[ w8_3 ] = neighbour( f8_3,sa,sb,7 );


%% Weight LBP
L1=w1+w2+w3+w4+w5+w6+w7+w8;

%% Weight LMep-1
L2=w1_1+w2_1+w3_1+w4_1+w5_1+w6_1+w7_1+w8_1;

%% Weight LMep-2
L3=w1_2+w2_2+w3_2+w4_2+w5_2+w6_2+w7_2+w8_2;

%% Weight LMep-3
L4=w1_3+w2_3+w3_3+w4_3+w5_3+w6_3+w7_3+w8_3;


%%
C1=reshape(L1,[],1);
C2=reshape(L2,[],1);
C3=reshape(L3,[],1);
C4=reshape(L4,[],1);

%%
D1=hist(C1,0:255);
D2=hist(C2,0:255);
D3=hist(C3,0:255);
D4=hist(C4,0:255);
```

```
D=[D1 D2 D3 D4];

% figure,
% bar(0:255,D);

histogram=D;
end
```

LBP_LMeP_Local

```
function [ histogram ] = lbp_lmep_local( I , ls , d  )

[sa,sb]=size(I); % 'I' is the Image     'ls' is local matrix size   'd' is no. of overlap
pixels
img=int32(I);

%% Forming Matrices for each matrix element
img1=img(2:sa-1,3:sb);
img2=img(1:sa-2,3:sb);
img3=img(1:sa-2,2:sb-1);
img4=img(1:sa-2,1:sb-2);
img5=img(2:sa-1,1:sb-2);
img6=img(3:sa,1:sb-2);
img7=img(3:sa,2:sb-1);
img8=img(3:sa,3:sb);
imgc=img(2:sa-1,2:sb-1);

%% Calculation for LBP

f1=img1-imgc;
f2=img2-imgc;
f3=img3-imgc;
f4=img4-imgc;
f5=img5-imgc;
f6=img6-imgc;
f7=img7-imgc;
f8=img8-imgc;

%% Calculation for LMep Index-1

f1_1=img2-img1;
f2_1=img3-img2;
f3_1=img4-img3;
f4_1=img5-img4;
f5_1=img6-img5;
f6_1=img7-img6;
f7_1=img8-img7;
```

```
f8_1=img1-img8;

%% Calculation for LMep Index-2

f1_2=img3-img1;
f2_2=img4-img2;
f3_2=img5-img3;
f4_2=img6-img4;
f5_2=img7-img5;
f6_2=img8-img6;
f7_2=img1-img7;
f8_2=img2-img8;

%% Calculation for LMep Index-3

f1_3=img4-img1;
f2_3=img5-img2;
f3_3=img6-img3;
f4_3=img7-img4;
f5_3=img8-img5;
f6_3=img1-img6;
f7_3=img2-img7;
f8_3=img3-img8;


%% Applying binary condition and Assigning weight for LBP

[ w1 ] = neighbour( f1,sa,sb,0 );
[ w2 ] = neighbour( f2,sa,sb,1 );
[ w3 ] = neighbour( f3,sa,sb,2 );
[ w4 ] = neighbour( f4,sa,sb,3 );
[ w5 ] = neighbour( f5,sa,sb,4 );
[ w6 ] = neighbour( f6,sa,sb,5 );
[ w7 ] = neighbour( f7,sa,sb,6 );
[ w8 ] = neighbour( f8,sa,sb,7 );

%% Applying binary condition and Assigning weight for LMep-1

[ w1_1 ] = neighbour( f1_1,sa,sb,0 );
[ w2_1 ] = neighbour( f2_1,sa,sb,1 );
[ w3_1 ] = neighbour( f3_1,sa,sb,2 );
[ w4_1 ] = neighbour( f4_1,sa,sb,3 );
[ w5_1 ] = neighbour( f5_1,sa,sb,4 );
[ w6_1 ] = neighbour( f6_1,sa,sb,5 );
[ w7_1 ] = neighbour( f7_1,sa,sb,6 );
[ w8_1 ] = neighbour( f8_1,sa,sb,7 );

%% Applying binary condition and Assigning weight for LMep-2
```

```
[ w1_2 ] = neighbour( f1_2,sa,sb,0 );
[ w2_2 ] = neighbour( f2_2,sa,sb,1 );
[ w3_2 ] = neighbour( f3_2,sa,sb,2 );
[ w4_2 ] = neighbour( f4_2,sa,sb,3 );
[ w5_2 ] = neighbour( f5_2,sa,sb,4 );
[ w6_2 ] = neighbour( f6_2,sa,sb,5 );
[ w7_2 ] = neighbour( f7_2,sa,sb,6 );
[ w8_2 ] = neighbour( f8_2,sa,sb,7 );

%% Applying binary condition and Assigning weight for LMep-3

[ w1_3 ] = neighbour( f1_3,sa,sb,0 );
[ w2_3 ] = neighbour( f2_3,sa,sb,1 );
[ w3_3 ] = neighbour( f3_3,sa,sb,2 );
[ w4_3 ] = neighbour( f4_3,sa,sb,3 );
[ w5_3 ] = neighbour( f5_3,sa,sb,4 );
[ w6_3 ] = neighbour( f6_3,sa,sb,5 );
[ w7_3 ] = neighbour( f7_3,sa,sb,6 );
[ w8_3 ] = neighbour( f8_3,sa,sb,7 );


%% Weight LBP
L1=w1+w2+w3+w4+w5+w6+w7+w8;

%% Weight LMep-1
L2=w1_1+w2_1+w3_1+w4_1+w5_1+w6_1+w7_1+w8_1;

%% Weight LMep-2
L3=w1_2+w2_2+w3_2+w4_2+w5_2+w6_2+w7_2+w8_2;

%% Weight LMep-3
L4=w1_3+w2_3+w3_3+w4_3+w5_3+w6_3+w7_3+w8_3;

%% Selecting the local method and the overlap size

i=1:ls-d:158-(ls-2)+1;
j=ls-2:ls-d:158;

len_i=length(i);
len_j=length(j);

imgl1=zeros(len_i,len_j,ls-2,ls-2);
imgl2=zeros(len_i,len_j,ls-2,ls-2);
imgl3=zeros(len_i,len_j,ls-2,ls-2);
imgl4=zeros(len_i,len_j,ls-2,ls-2);

for a=1:length(i)
    for b=1:length(j)
        imgl1(a,b,:,:)=L1(i(a):j(a),i(b):j(b));
```

```
        imgl2(a,b,:,:)=L2(i(a):j(a),i(b):j(b));
        imgl3(a,b,:,:)=L3(i(a):j(a),i(b):j(b));
        imgl4(a,b,:,:)=L4(i(a):j(a),i(b):j(b));
    end
end

%% Reshaping the matrices and forming local histograms
hist_temp=zeros(len_i,len_j,1024);
for a=1:len_i
    for b=1:len_j
        TV1=imgl1(a,b,:,:);
        TV2=imgl2(a,b,:,:);
        TV3=imgl3(a,b,:,:);
        TV4=imgl4(a,b,:,:);
        C1=reshape(TV1,[],1);
        C2=reshape(TV2,[],1);
        C3=reshape(TV3,[],1);
        C4=reshape(TV4,[],1);
        D1=hist(C1,0:255);
        D2=hist(C2,0:255);
        D3=hist(C3,0:255);
        D4=hist(C4,0:255);
        D=[D1 D2 D3 D4];
        hist_temp(a,b,:)=D;
    end
end

histogram=hist_temp;

end
```

Neighbour Function

```
function [ w ] = neighbour( g, sa , sb, n )
% The neighbour function point out all the neighbouring
%  pixels corresponding to the centre pixel.

for i=(1:sa-2)
        for j=(1:sb-2)
            if g(i,j)>=0
                g(i,j)=1;
            else
                g(i,j)=0;
            end
        end
end
b = g;
w=((2^n)*b);
end
```

Check Function

```
clc
clear all
%% Image matching and accuracy calculation

m= input('Please enter the no. of people to check : ');
k=input('Please enter the no. of images to be considered for output :');


choice=input('Which case do you want to check \n 1.Global \n 2.Local : ');

fprintf('\n\nPlease wait . . . Computing the code precision.\n\n')

tic
if (choice==1)
   load ('feature_database_global_9_F.mat','feature');       % loading the database
   [n,~] = size(feature);
   p=zeros(1,m);
   for i=1:m
      [precision] = checker_global( i, feature, n, k);
      p(1,i)=precision;
   end
   s=sum(p);
   Pavg=s./m;
   fprintf('\n\nThe average precision of the "global" algorithm among %d persons
is = %f percent \n',m,Pavg);

elseif (choice==2)

   load ('feature_database_local_7a_F.mat','feature');       % loading the database
   [n,~] = size(feature);

   ls=input('Please enter the size of the local matrix:');
   d=input('Please enter the number of overlap column:');
   p=zeros(1,m);
   for i=1:m
      [ precision ] = checker_local( i, ls, n, feature, d, k );
      p(1,i)=precision;
   end
   s=sum(p);
   Pavg=s./m;
   fprintf('\n\nThe average precision of the "local" algorithm among %d persons is
= %f percent \n',m,Pavg);
```

```
toc
end
```

Checker Function

```
function [ precision ] = checker( p, feature, n, k )

filename=strcat('T_PolyU_',int2str(p),'_F_1','.bmp');
I=imread(filename);
[ histogram ] = lbp_lmep(I);
C=histogram;
listf=zeros(1,n);
for i = 1:n
    num = (C)-(feature{i,1});
    denom = 1+abs(C)+abs(feature{i,1});
    dist_d1 = abs(num.^1)./abs(denom.^1);
    sum_col = sum(dist_d1);
    listf(1,i) = sum_col;
end

[dist_sort,idx] = sort(listf);

acc=0;


for j=1:k
    for o=1:9
        if (idx(1,j)==(((p-1)*9)+o))
            acc=acc+1;
            break
        end
    end
end

inacc=k-acc;

precision=round(100*(acc./(acc+inacc)));

end
```

Checker_global

```
function [ precision ] = checker_global( p, feature, n, k )

filename=strcat('C_PolyU_',int2str(p),'_F_7','.bmp');
I=imread(filename);
```

```
[ histogram ] = lbp_lmep(I);
C=histogram;
listf=zeros(1,n);
for i = 1:n
    num = (C)-(feature{i,1});
    denom = 1+abs(C)+abs(feature{i,1});
    dist_d1 = abs(num.^1)./abs(denom.^1);
    sum_col = sum(dist_d1);
    listf(1,i) = sum_col;
end

[dist_sort,idx] = sort(listf);

acc=0;
number=3;

for j=1:k
    for o=1:number
        if (idx(1,j)==(((p-1)*number)+o))
            acc=acc+1;
            break
        end
    end
end

inacc=k-acc;

precision=round(100*(acc./(acc+inacc)));

end
```

Checker_local function

```
function [ precision ] = checker_local( p, ls, n, feature, d, k )

filename=strcat('C_PolyU_',int2str(p),'_F_1','.bmp');
I=imread(filename);

[ histogram ] = lbp_lmep_local( I , ls , d );
C=histogram;
listf=zeros(1,n);

for i = 1:n
    tmp=feature{i,1};
%     tmp1=cell2mat(tmp);
    num = (C)-(tmp);
```

```
    denom = 1+abs(C)+abs(tmp);
    dist_d1 = abs(num.^1)./abs(denom.^1);
    L1=reshape(dist_d1,[],1);
    sum_col = sum(L1);
    listf(1,i) = sum_col;
end

[dist_sort,idx] = sort(listf);

acc=0;

number=7;

for j=1:k
    for o=1:number
        if (idx(1,j)==(((p-1)*number)+o))
            acc=acc+1;
            break
        end
    end
end

inacc=k-acc;

precision=round(100*(acc./(acc+inacc)));

end
```

References:

"A Survey of Palmprint Recognition." Accessed May 14, 2015. http://www.sciencedirect.com/science/article/pii/S0031320309000399.

"BBC News | SCI/TECH | Doubt Cast on Fingerprint Security." Accessed May 14, 2015. http://news.bbc.co.uk/2/hi/science/nature/1991517.stm.

"Biometric Personal Authentication Using Keystroke Dynamics: A Review." Accessed May 14, 2015. http://www.sciencedirect.com/science/article/pii/S156849461000205X.

"Biometrics.pdf," n.d.

"IEEE Xplore Abstract - Dominant Local Binary Patterns for Texture Classification." Accessed May 14, 2015. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4808422.

"IEEE Xplore Abstract - Local Mesh Patterns Versus Local Binary Patterns: Biomedical Image Indexing and Retrieval." Accessed May 14, 2015. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6654267.

"Palm Print Recognition.pdf," n.d.

"Palmprint Verification Using Binary Orientation Co-Occurrence Vector." Accessed May 14, 2015. http://www.sciencedirect.com/science/article/pii/S0167865509001196.

"Palm Print - Wikipedia, the Free Encyclopedia." Accessed May 14, 2015. http://en.wikipedia.org/wiki/Palm_print.