Palmprint based Recognition System robust to partial occlusion

Devendra Kumar Arya (Y5827165)

Under the supervision of
Prof. Phalguni Gupta

Computer Science and Engineering
Indian Institute of Technology, Kanpur

August 10, 2010

Overview

Section

- Biometric based recognition is the technique of automatic identification or verification of a person with the help of his physiological or behavioral characteristics
  - Physiological characteristics include finger scans, hand scans, retinal scans, iris, face, palmprint scans
  - Behavioral characteristics include voice scans, keystroke scans (typing rythm) and signature scans
- Biometric based recognition has several advantages over traditional methods: token based approach and knowledge based approach
  - A biometric trait cannot be easily transferred, forgotten or lost
  - The rightful owner of the biometric template can be easily identified
  - It is difficult to duplicate a biometric trait

a typical Biometric System Architechture

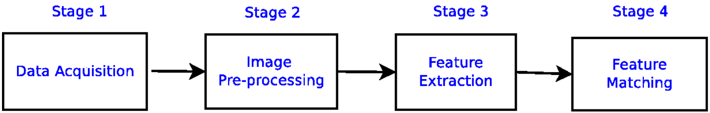


Figure 1: Biometric System Architechture

Properties of an Effective Biometric trait

Universality: Every person should have the biometric characteristic

Uniqueness: No two persons should be the same in terms of biometric characteristic

Permanence: The biometric characteristic should be invariant over time

Collectability: The biometric characteristic should be measurable with some sensing device

Acceptability: Population should have no objection to providing the biometric

Performance: It refers to achievable accuracy and speed

Circumvention: A measure of the way to forge the system using fraudulent methods

Why Palmprint?

- A palmprint based biometric system has high user acceptability
- Even the palmprints of mono-zygotic twins are distinct
- Palmprint is regarded as one of the most stable, unique and reliable characteristics
- Palm exhibits very little change in its structure due to aging
- High recognition rates can be obtained by utilizing even low resolution palmprint images
- People (general public) consider to provide their palmprints as non-intrusive

Motivation

- Biometric based recognition systems (face, iris, ear, palmprint) use an image obtained from an acqusition device for person recognition
- In many cases, it has been observed that the acquired images are partially occluded i.e. they have a portion of the image as unavailable
    - the person from whom the sample is being collected has injuries or is physically challenged
    - misallignment of the acquisition device with respect to the trait while collecting the sample
- In such boundary cases:
    - system entirely rejects/accepts the person (this happens when the system is not robust to partial occlusions)
    - Such cases are handled manually

Problem Definition

To design an efficient palmprint based recognition system which is robust to partial occlusion

Desirable features of the system:

- It should perform well with low resolution palmprint images so that even a low cost scanner can be used to achieve efficient performances
- It should perform equally well with acquisition devices which have constraints using pegs as well as those which are pegs-free so that physically challenged people can also be enrolled and recognized

Section

Identification

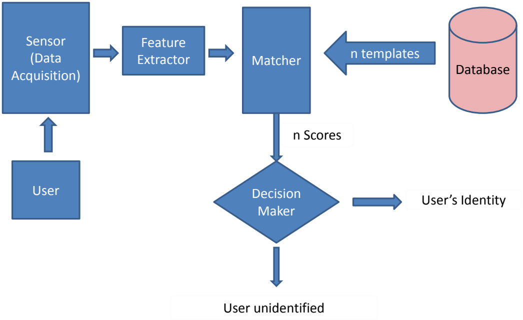


Figure 2: Identification

Verification

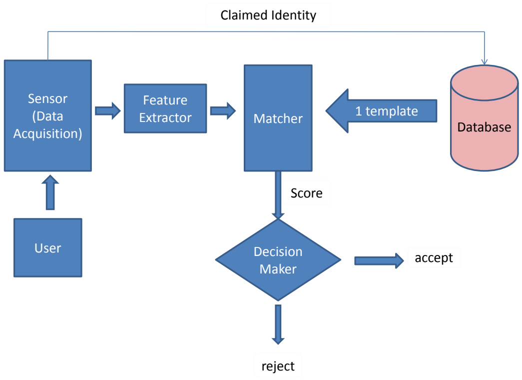


Figure 3: Verification

Palm features

Approaches for palmprint based recognition can be classified into two categories based on palm features: Structural Palm features, Statistical Palm features

Structural Palm features: These features are based on hand geometry, minutiae points, wrinkles, principle lines, delta points etc.

Statistical Palm features: fourier transform, Discrete Cosine Transform, K-L transform, 2 D Gabor phase encoding, SIFT, SURF operators

Force Field Transformation [Hurley *et. al.*]

Given an image f: x $\times$ y $\rightarrow P$ where P is a value in [0, 255] gray scale intensity levels. Each pixel is assumed to generate a spherically symmetrical force field. [1]
The force $\mathbf{F}_i(\mathbf{r})$ exerted on a pixel of unit intensity at the pixel location with position vector $\mathbf{r}$ by a remote pixel with position vector $\mathbf{r}_i$ and pixel intensity $\mathrm{P}(\mathbf{r}_i)$ is given by

$$\mathbf{F}_i(\mathbf{r}) = P(\mathbf{r}_i)\frac{\mathbf{r}_i - \mathbf{r}}{\left|\mathbf{r}_i - \mathbf{r}\right|^3} \qquad (1)$$

[1] D. J. Hurley, M. S. Nixon, J. N. Carter. Force Field Functionals for Image Feature Extraction. *Image and Vision Computing*, 20(5-6):311-317, 2002

Contd..

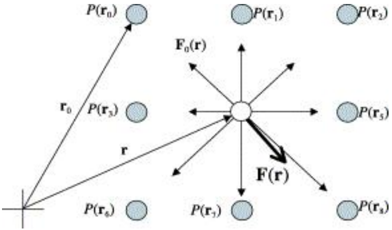


Figure 4: Direction of forces exerted by a pixel of unit intensity located at position vector r due to eight pixels (with position vectors: $\mathbf{r}_0$ $\mathbf{r}_1$ $\mathbf{r}_2$ $\mathbf{r}_3$ $\mathbf{r}_5$ $\mathbf{r}_6$ $\mathbf{r}_7$) in its neighbourhood. Source [1]

Local Structure Tensor

- Palmprint image is rich in texture [2]
- Local orientation is a very big component of the texture present in an image
- Local Structure tensor (LST) is a tensor representation of this orientation in multi-dimensional signals [3]
- Local Structure Tensor (LST) is widely accepted to provide a compact representation of this local orientation [4]

Local Structure Tensor

- The computation of the local structure tensor components are given by the following formulation:

$$\mathbf{T}(\mathbf{x}_0) = \frac{1}{4\pi^2} \int_{\mathbf{x}\,\epsilon\,\Omega(\mathrm{x}_0)} \frac{\partial f}{\partial x_i} \frac{\partial f}{\partial y_j}\, \mathrm{d}\mathbf{x} \tag{2}$$

  where $\Omega(x_0)$ is a local neighbourhood of the function f($\mathbf{x}$) around the point $\mathbf{x}_0$ where the structure tensor is going to be estimated.

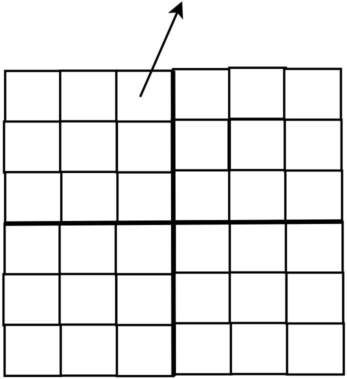
- Discrete version of this equation is:

$$\mathbf{T}(\mathbf{x}_0) = G_\sigma(\mathbf{x}_0) * (\triangledown f(\mathbf{x}_0) \triangledown^T f(\mathbf{x}_0)) \tag{3}$$

  where G_σ is a Gaussian filter of standard deviation σ

LST matrix of a pixel

$$\begin{pmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{pmatrix}$$



Contd..

$$G(x, y) = \frac{1}{2\pi\sigma} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{4}$$

$$G_x = \frac{\partial}{\partial x} G \tag{5}$$

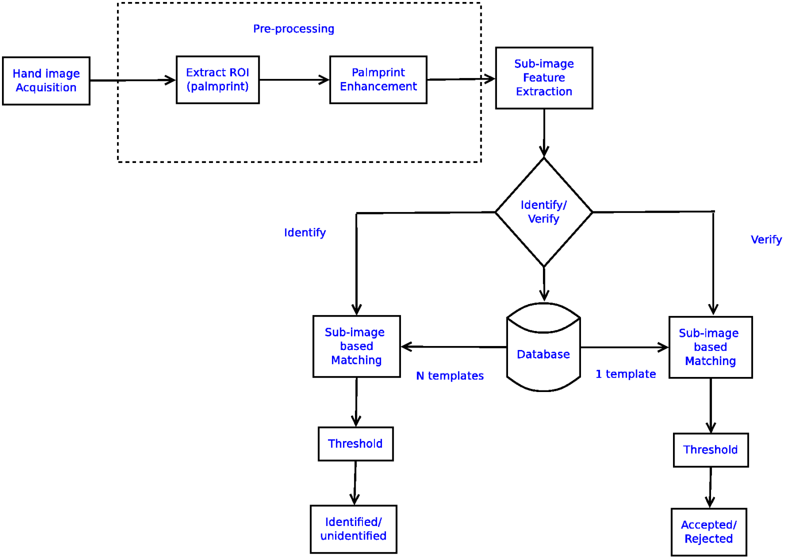
$$G_y = \frac{\partial}{\partial y} G \tag{6}$$

$$I * G_x = I_x \tag{7}$$

$$I * G_y = I_y \tag{8}$$

Section

Flow Diagram



Region of Interest (ROI) Extraction

Hand image obtained from the acquisition device is first binarized and then hand contour is obtained using contour tracing algorithm [5].

(a) (b) (c)

Figure 7: (a) Hand image (b) Binarized image (c) Hand contour

Contd..

- Fingertip and valley co-ordinates are obtained from local maxima and minima on the contour image
- Two reference points, (1) V_1, the valley point between ring finger and little finger, and (2) V_2, the valley point between forefinger and middle finger, are taken on the contour
- The two reference points are then connected with a line as shown in the figure

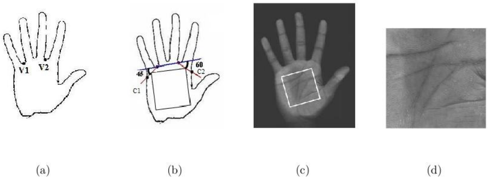


Figure 8: (a) Hand contour showing reference points: V_1, V_2. (b) ROI on hand contour. (c) ROI shown in gray-scale image. (d) Extracted ROI (palmprint).

Contd..

- Two lines, L_1 and L_2, are drawn at angles 45 degree and 60 degree at points V_1 and V_2 respectively
- Let C_1 and C_2 be the 2 points where L_1 and L_2 meet the hand boundary
- The two midpoints M_1 and M_2 of the two line segment joining V_1 and C_1, and, V_2 and C_2, respectively are obtained
- With the line $M_1 - M_2$ as a side, a square is drawn. This square region is the Region Of Interest (ROI) and is termed as *palmprint*

Palmprint Enhancement

- The palmprint obtained has non-uniform brightness due to non-uniform reflection of light from the surface of the palm having a small curvature

Palmprint Enhancement

- The palmprint is divided into 32×32 sub-blocks and mean for each sub-block is computed. This provides an estimate of the background reflection
- The estimated background reflection is expanded using bicubic interpolation to original size of palmprint image
- Estimated reflection is subtracted from the original image. This compensates the variation of brightness across the image, as a result of which, a uniform brightness or illumination corrected palmprint image is obtained
- Histogram equalization is then performed on 64×64 blocks for contrast enhancement of the uniform brightness or illumination corrected image to obtain enhanced palmprint

Wang *et. al.* [6] have used a similar procedure for texture enhancement in iris images

Contd..

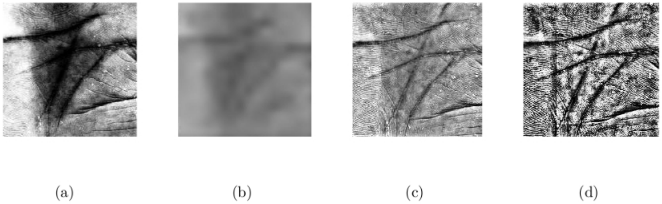


Figure 9: (a) Extracted palmprint. (b) Estimated coarse reflection. (c) Uniform brightness palmprint image. (d) Enhanced palmprint image.

Sub-image based Feature Extraction

- The enhanced palmprint P (of size $n \times n$) is divided into $m \times m$ subimages
- Features are extracted from each sub-image

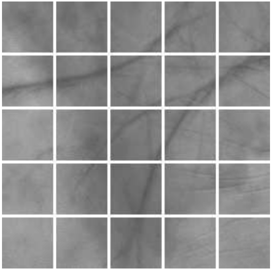
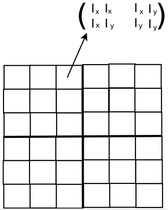


Figure 10: A palmprint image partitioned into $5 \times 5 = 25$ sub-images

Feature Extraction using Local Structure Tensor

1. The original enhanced palmprint image (let us say, image I with dimensions n $\times$ n)is divided into m $\times$ m sub-images
2. For each sub-image I_{sub}, obtain the Gaussian based filters, G_x and G_y and convolve the image I_{sub} to obtain the gradients $(I_{sub})_x$ and $(I_{sub})_y$ respectively
3. For each pixel p_i that belongs to sub-image I_{sub}, the structure tensor matrix T_{p_i} is obtained:

$$T_{p_i} = \begin{bmatrix} (I_{sub})_x(p_i)(I_{sub})_x(p_i) & (I_{sub})_x(p_i)(I_{sub})_y(p_i) \\ (I_{sub})_x(p_i)(I_{sub})_y(p_i) & (I_{sub})_y(p_i)(I_{sub})_y(p_i) \end{bmatrix} \quad (9)$$



Feature Extraction using Local Structure Tensor

$$T_{sub} = \frac{1}{\#I_{sub}} \begin{bmatrix} \sum_{p_i \epsilon I_{sub}} (I_{sub})_x(p_i)(I_{sub})_x(p_i) & \sum_{p_i \epsilon I_{sub}} (I_{sub})_x(p_i)(I_{sub})_y(p_i) \\ \sum_{p_i \epsilon I_{sub}} (I_{sub})_x(p_i)(I_{sub})_y(p_i) & \sum_{p_i \epsilon I_{sub}} (I_{sub})_y(p_i)(I_{sub})_y(p_i) \end{bmatrix} \tag{10}$$

Feature Extraction using Local Structure Tensor

1. The original enhanced palmprint image (let us say, image I with dimensions n $\times$ n)is divided into m $\times$ m sub-images
2. For each sub-image I_{sub}, obtain the Gaussian based filters, G_x and G_y and convolve the image I_{sub} to obtain the gradients $(I_{sub})_x$ and $(I_{sub})_y$ respectively
3. For each pixel p_i that belongs to sub-image I_{sub}, the structure tensor matrix T_{p_i} is obtained:

$$T_{p_i} = \begin{bmatrix} (I_{sub})_x(p_i)(I_{sub})_x(p_i) & (I_{sub})_x(p_i)(I_{sub})_y(p_i) \\ (I_{sub})_x(p_i)(I_{sub})_y(p_i) & (I_{sub})_y(p_i)(I_{sub})_y(p_i) \end{bmatrix} \tag{11}$$

4. Obtain the mean of T_{p_i}. Let the 2×2 mean matrix be denoted by T_{sub}

Contd..

- Eigen decomposition is performed on T_{sub} to obtain two principal components
- Let e_1, e_2 be the two eigen vectors and λ_1, λ_2 the corresponding eigen values
- The dominant eigen vector *i.e.* the eigen vector corresponding to the higher eigen value which stores more information regarding the change in the gradient is taken and is stored as the feature vector for the sub-image
- Similarly, a feature vector for each sub-image is obtained
- The feature matrix for the whole image I is obtained by concatenation of individual feature vectors from each sub-image

Contd..

$$\textbf{Feature matrix}_I = \begin{bmatrix} f_{1,1} & f_{1,2} & \cdots & f_{1,m} \\ f_{2,1} & f_{2,2} & \cdots & f_{2,m} \\ \vdots & \vdots & \cdots & \vdots \\ f_{m,1} & f_{m,2} & \cdots & f_{m,m} \end{bmatrix} \quad (12)$$

where $f_{i,j}$ is the feature vector for the $(i,j)^{th}$ sub-image of the palmprint image I

Contd...

Algorithm 1: LST_Based_FEAT_EXTR(I, m)

Input: Given a gray scale image I of dimension $n \times n$ and a number m where $m \times m$ is the number of sub-images in which I is divided.

Output: A feature matrix **Feature matrix**$_I$ $= \begin{bmatrix} f_{1,1} & f_{1,2} & \cdots & f_{1,m} \\ f_{2,1} & f_{2,2} & \cdots & f_{2,m} \\ \vdots & \vdots & \cdots & \vdots \\ f_{m,1} & f_{m,2} & \cdots & f_{m,m} \end{bmatrix}$

$r \Leftarrow n \bmod m$
if $r \neq 0$ **then**
  if $r < \lfloor \frac{m}{10} \rfloor$ **then**
    remove the last r rows and columns
  else
    Pad the palmprint with $m - r$ rows and $m - r$ columns of 0's
    $m \leftarrow m + 1$
  end if
end if
Divide the palmprint into $m \times m$ sub-images
Let I_{ij} denote the $(i,j)^{th}$ sub-image of the input image I
for $i = 1$ **to** m **do**
  for $j = 1$ **to** m **do**
    Obtain the Gaussian based filters, G_x and G_y for I_{ij}
    Convolve I_{ij} with G_x and G_y to obtain gradient images $(I_{ij})_x$ and $(I_{ij})_y$ respectively
    for each pixel $p_a \; \epsilon \; I_{ij}$ **do**
      $T_{p_a} \leftarrow \begin{bmatrix} (I_{ij})_x(p_a) \times (I_{ij})_x(p_a) & (I_{ij})_x(p_a) \times (I_{ij})_y(p_a) \\ (I_{ij})_x(p_a) \times (I_{ij})_y(p_a) & (I_{ij})_y(p_a) \times (I_{ij})_y(p_a) \end{bmatrix}$
    end for
    $T_{ij} \leftarrow \frac{1}{\# I_{ij}} \begin{bmatrix} \sum_{p_a \epsilon I_{ij}} T_{p_a}(1,1) & \sum_{p_a \epsilon I_{ij}} T_{p_a}(1,2) \\ \sum_{p_a \epsilon I_{ij}} T_{p_a}(2,1) & \sum_{p_a \epsilon I_{ij}} T_{p_a}(2,2) \end{bmatrix}$
    $(e_1, e_2, \lambda_1, \lambda_2) \leftarrow EigenDecomposition(T_{ij})$

Contd...

```
        if λ1 > λ2 then
            f_{i,j} ← e1
        else
            f_{i,j} ← e2
        end if
    end for
end for
return f
```

Force Field based Filter

- A force field based filter using the idea in [1] is proposed
- Force Field based filter computes the impact of neighbourhood pixels on the central pixel
- The Force Field based filter uses a $k \times k$ window which is passed with its centre falling on each pixel of the image on turn

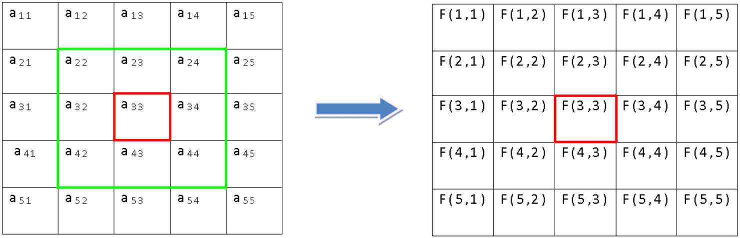


Figure 13: Force Field filter

Contd..

- Let the pixel at the centre of the $k \times k$ window be r_{centre}
- The effect of each pixel in the vicinity of r_{centre} falling under the window is computed using the following formula:

$$\mathbf{F}_i(\mathbf{r}_{centre}) = P(\mathbf{r}_i)\frac{\mathbf{r}_i - \mathbf{r}_{centre}}{|\mathbf{r}_i - \mathbf{r}_{centre}|^3} \times P(r_{centre}) \tag{13}$$

$$\mathbf{F}(\mathbf{r}_{centre}) = \sum_i \mathbf{F}_i(\mathbf{r}_{centre}) \tag{14}$$

where $P(\mathbf{r}_i)$ denotes the pixel value at $\mathbf{r}_i$ and i varies so as to include every pixel lying in the $k \times k$ window.

Contd..

- a force F(i,j) for every pixel of an image considering the pulling effect of those pixels that lie in the $k \times k$ window that is applied to it can be obtained

$$F(i,j) = a\hat{i} + b\hat{j} \tag{15}$$

- The direction of this force can be obtained by:

$$\theta(\mathbf{r}) = tan^{-1}(\frac{b}{a}) \tag{16}$$

- The direction θ for a pixel holds the information telling the net pulling effect on a pixel due to its neighbourhood pixels
- The value θ is the orientation of a pixel that it experiences due to all pixels falling in the $k \times k$ window combined

dominant Orientation pixels of a sub-image

- In Algorithm 1, LST matrix is computed for each pixel. Computationwise, this is an expensive process
- A method in which LST matrix is evaluated only for certain special pixels is more desirable
- For each sub-image, a set of dominant pixels D_{sub} is evaluated. These pixels are also referred to as *dominant orientation pixels*
- LST matrix would then be evaluated only for these pixels

Computation of D_{sub} the set of dominant Orientation pixels of a sub-image

1. For an enhanced palmprint image $[I]$, the corresponding filtered image $[F]$ using force field based filter described earlier is obtained $F_{ij} = a\hat{i} + b\hat{j}$
2. The direction matrix $[\theta]$ which holds the orientation of each pixel with respect to force field in a window of size k $\times$ k is obtained $\theta_{ij} = tan^{-1}(\frac{b}{a})$
3. Divide the matrix $[\theta]$ into $m \times m$ cells where m corresponds to the number of sub-images features are to be extracted from
4. For each cell, the orientation values which ϵ [0, 360] are distributed into 8 bins: $\theta_{min} - \theta_1$, $\theta_1 - \theta_2$, $\theta_2 - \theta_3$, $\cdots$, $\theta_7 - \theta_{max}$ where $\theta_i = \theta_{min} + i * (\frac{\theta_{max} - \theta_{min}}{8})$
5. A distribution histogram is plotted for the 8 bins. The bin with the highest frequency is considered as the dominant orientation of the cell. The pixels that are falling into this bin are collected and are said to belong to the set of dominant pixels D_{sub}

Feature Extraction using Local Structure Tensor: Algorithm 2

1. The original enhanced palmprint image (let us say, image I with dimensions n $\times$ n) is divided into m $\times$ m sub-images
2. For each sub-image I_{sub}, obtain the Gaussian based filters, G_x and G_y and convolve the image I_{sub} to obtain the gradients $(I_{sub})_x$ and $(I_{sub})_y$ respectively.
3. For each pixel p_i that belongs to sub-image I_{sub} and the set D_{sub} for that sub-image, the structure tensor matrix T_{p_i} is obtained:

$$T_{p_i} = \begin{bmatrix} (I_{sub})_x(p_i)(I_{sub})_x(p_i) & (I_{sub})_x(p_i)(I_{sub})_y(p_i) \\ (I_{sub})_x(p_i)(I_{sub})_y(p_i) & (I_{sub})_y(p_i)(I_{sub})_y(p_i) \end{bmatrix} \tag{17}$$

4. Obtain the mean of T_{p_i}. Let the 2×2 mean matrix be denoted by T_{sub}.

Feature Extraction using Local Structure Tensor: Algorithm 2 Contd..

$$T_{sub} = \frac{1}{\#D_{sub}} \begin{bmatrix} \sum\limits_{p_i \epsilon I_{sub} \& D_{sub}} (I_{sub})_x(p_i)(I_{sub})_x(p_i) & \sum\limits_{p_i \epsilon I_{sub} \& D_{sub}} (I_{sub})_x(p_i)(I_{sub})_y(p_i) \\ \sum\limits_{p_i \epsilon I_{sub} \& D_{sub}} (I_{sub})_x(p_i)(I_{sub})_y(p_i) & \sum\limits_{p_i \epsilon I_{sub} \& D_{sub}} (I_{sub})_y(p_i)(I_{sub})_y(p_i) \end{bmatrix} \tag{18}$$

Contd..

- Eigen decomposition is performed on T_{sub} to obtain two principal components
- Let e_1, e_2 be the two eigen vectors and λ_1, λ_2 the corresponding eigen values
- The dominant eigen vector *i.e.* the eigen vector corresponding to the higher eigen value which stores more information regarding the change in the gradient is taken and is stored as the feature vector for the sub-image
- Similarly, a feature vector for each sub-image is obtained
- The feature matrix for the whole image I is obtained by concatenation of individual feature vectors from each sub-image

Contd..

$$\textbf{Feature matrix}_I = \begin{bmatrix} f_{1,1} & f_{1,2} & \cdots & f_{1,m} \\ f_{2,1} & f_{2,2} & \cdots & f_{2,m} \\ \vdots & \vdots & \cdots & \vdots \\ f_{m,1} & f_{m,2} & \cdots & f_{m,m} \end{bmatrix} \quad (19)$$

where $f_{i,j}$ is the feature vector for the $(i,j)^{th}$ sub-image of the palmprint image I

Contd...

Algorithm 2: LST_DOMINANT_PIXELS_FEAT_EXTR(I, θ, m)

Input: Given a gray scale image I of dimension $n \times n$, orientation matrix $[\theta]$ of dimension $n \times n$ and a number m where $m \times m$ is the number of sub-images in which I is divided.

Output: A feature matrix $\textbf{Feature matrix}_I = \begin{bmatrix} f_{1,1} & f_{1,2} & \cdots & f_{1,m} \\ f_{2,1} & f_{2,2} & \cdots & f_{2,m} \\ \vdots & \vdots & \cdots & \vdots \\ f_{m,1} & f_{m,2} & \cdots & f_{m,m} \end{bmatrix}$

$r \Leftarrow n \bmod m$
if $r \neq 0$ **then**
  if $r < \lfloor \frac{m}{10} \rfloor$ **then**
    remove the last r rows and columns
  else
    Pad the palmprint with $m - r$ rows and $m - r$ columns of 0's
    $m \leftarrow m + 1$
  end if
end if
Divide the palmprint into $m \times m$ sub-images
Divide the matrix $[\theta]$ into $m \times m$ cells
Let I_{ij} denote the $(i, j)^{th}$ sub-image of the input image I
Let θ_{ij} denote the $(i, j)^{th}$ cell of the orientation matrix $[\theta]$
for $i = 1$ **to** m **do**
  for $j = 1$ **to** m **do**
    Obtain the Gaussian based filters, G_x and G_y for I_{ij}
    Convolve I_{ij} with G_x and G_y to obtain gradient images $(I_{ij})_x$ and $(I_{ij})_y$ respectively
    $\theta_{min} \leftarrow \min \{\theta_{ij}\}$
    $\theta_{max} \leftarrow \max \{\theta_{ij}\}$
    $\theta_i = \theta_{min} + i * (\frac{\theta_{max} - \theta_{min}}{8})$.
    Distribute the orientation values of θ_{ij} into 8 equal bins: $\theta_{min} - \theta_1$, $\theta_1 - \theta_2$, $\theta_2 - \theta_3$, $\ldots$, $\theta_7 - \theta_{max}$

Contd...

The bin with the highest frequency is considered as the dominant orientation of the cell θ_{ij}. The pixels that are falling into this bin are collected and are said to belong to the set of dominant pixels D_{ij}

for each pixel $p_a \epsilon I_{ij}$ and D_{ij} **do**

$$T_{p_a} \leftarrow \begin{bmatrix} (I_{ij})_x(p_a) \times (I_{ij})_x(p_a) & (I_{ij})_x(p_a) \times (I_{ij})_y(p_a) \\ (I_{ij})_x(p_a) \times (I_{ij})_y(p_a) & (I_{ij})_y(p_a) \times (I_{ij})_y(p_a) \end{bmatrix}$$

end for

$$T_{ij} \leftarrow \frac{1}{\#D_{ij}} \begin{bmatrix} \sum\limits_{p_a \epsilon I_{ij} \& D_{ij}} T_{p_a}(1,1) & \sum\limits_{p_a \epsilon I_{ij} \& D_{ij}} T_{p_a}(1,2) \\ \sum\limits_{p_a \epsilon I_{ij} \& D_{ij}} T_{p_a}(2,1) & \sum\limits_{p_a \epsilon I_{ij} \& D_{ij}} T_{p_a}(2,2) \end{bmatrix}$$

$(e_1, e_2, \lambda_1, \lambda_2) \leftarrow EigenDecomposition(T_{ij})$

if $\lambda_1 > \lambda_2$ **then**

  $f_{i,j} \leftarrow e_1$

else

  $f_{i,j} \leftarrow e_2$

end if

end for

end for

return f

Sub-image based Matching

- The difference between feature matrix of live palmprint and enrolled palmprint is calculated and is termed as *matching score*
- Depending on the value of the score with respect to a pre-defined threshold, a decision is made and the user is either declared as genuine or impostor
- Let L and E be the feature matrices of the live palmprint (LI) and enrolled palmprint (EI) partitioned into $m \times m$ sub-images respectively.

$$L = \begin{bmatrix} l_{1,1} & l_{1,2} & \cdots & l_{1,m} \\ l_{2,1} & l_{2,2} & \cdots & l_{2,m} \\ \vdots & \vdots & \ldots & \vdots \\ l_{m,1} & l_{m,2} & \cdots & l_{m,m} \end{bmatrix} E = \begin{bmatrix} e_{1,1} & e_{1,2} & \cdots & e_{1,m} \\ e_{2,1} & e_{2,2} & \cdots & e_{2,m} \\ \vdots & \vdots & \ldots & \vdots \\ e_{m,1} & e_{m,2} & \cdots & e_{m,m} \end{bmatrix} \quad (20)$$

Contd..

The live palmprint is said to have matched enrolled palmprint if:

$$\frac{\sum_{i=1}^{m}\sum_{j=1}^{m}||l_{i,j}-e_{i,j}||}{m\times m} < Th \tag{21}$$

where Th is a pre-defined threshold, $||l_{i,j}-e_{i,j}||$ denotes the square of the modulus of the difference vector between $(i,j)^{th}$ sub-image of the live and enrolled palmprint

Robustness to Occlusion

- Entropy is a statistical measure of randomness that characterises the texture of an image [7]
- Based on the entropy of a sub-image, it can either be classified as occluded or non-occluded
- Let LI denote the live palmprint. Then the entropy of a subimage (i,j) of the palmprint $LI_{i,j}$ is given by

$$-\sum_{x=0}^{255} LI_{i,j}^{x} * \log_2(LI_{i,j}^{x}) \tag{22}$$

where $LI_{i,j}^{x}$ denotes the number of pixels of intensity x in the sub-image $LI_{i,j}$

Contd..

Now, the live palmprint is said to have matched enrolled palmprint if:

$$\frac{\sum_{i=1}^{m}\sum_{j=1}^{m} ||l_{i,j} - e_{i,j}|| \times Occ_{i,j}}{m \times m} < Th \tag{23}$$

where $Occ_{i,j}$ is defined as

$$Occ_{i,j} = \begin{cases} 1 & if -\sum_{x=0}^{255} LI_{i,j}^{x} * \log_2(LI_{i,j}^{x}) \leq OccTh \\ 0 & otherwise \end{cases} \tag{24}$$

where $OccTh$ is pre-defined randomness threshold

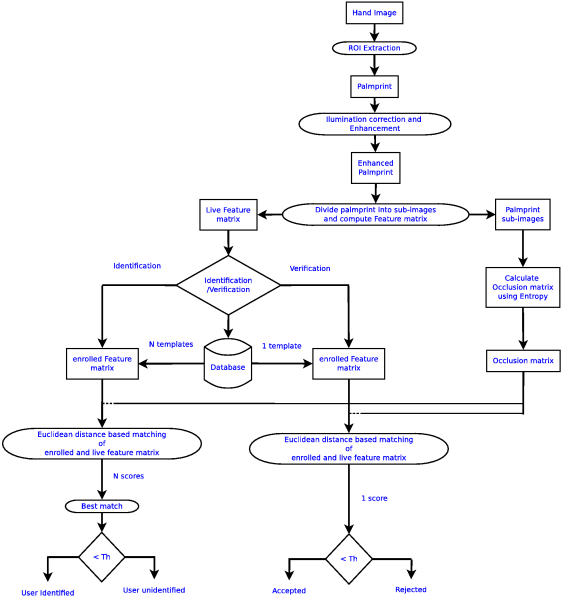


Figure 15: Proposed System

Section

Following metrics are used for measuring the performance of the proposed system:

FAR (False Acceptance Rate): percentage of invalid matches

GAR (Genuine Acceptance Rate): percentage of genuine matches

FRR (False Rejection Rate): percentage of genuine users rejected. $FRR = 100 - GAR$.

EER (Equal Error Rate): the rate at which FAR is equal to FRR. This parameter is used to compare two systems. Lower the EER better is the system.

ROC curve: A system's performance can be changed by changing matching threshold. $\mathrm{ROC}(T) = (\mathrm{FAR}(T), \mathrm{GAR}(T))$ where T is the threshold is a curve that tells the trade-off between FAR and GAR as the matching threshold is varied

Databases

- IITK Palm Database: 183 different subjects, 3 images per subject $\Rightarrow$ 549 hand images
- CASIA: 602 different subjects, 8 images per subject $\Rightarrow$ 5238 hand images
- PolyU: 386 different subjects, 16 images per subject $\Rightarrow$ 7752 hand images

IITK palm database

IITK					
	4×4	8×8	10×10	12×12	16×16
Accuracy (%)	75.57	80.05	77.68	**80.53**	80.56
FAR (%)	20.56	7.66	3.84	**11.30**	9.92
FRR (%)	28.28	32.23	40.78	**27.63**	28.94
EER (%)	25.68	22.36	22.36	**22.41**	22.85

Table 1: Performance of proposed system: Algorithm 1 on IITK database

CASIA

CASIA					
	4×4	8×8	10×10	12×12	16×16
Accuracy (%)	89.51	95.35	96.91	**97.03**	94.46
FAR (%)	9.07	1.94	1.56	**1.51**	1.99
FRR (%)	11.89	7.34	4.60	**4.42**	9.09
EER (%)	10.74	5.69	3.30	**3.49**	7.24

Table 2: Performance of proposed system: Algorithm 1 on CASIA database

PolyU

PolyU					
	4×4	8×8	10×10	12×12	16×16
Accuracy (%)	81.97	95.49	98.18	**98.38**	98.10
FAR (%)	15.78	3.74	1.09	**0.80**	0.98
FRR (%)	20.27	5.27	2.55	**2.42**	2.81
EER (%)	18.31	4.77	1.99	**1.88**	2.08

Table 3: Performance of proposed system: Algorithm 1 on PolyU database

IITK: ROC curve

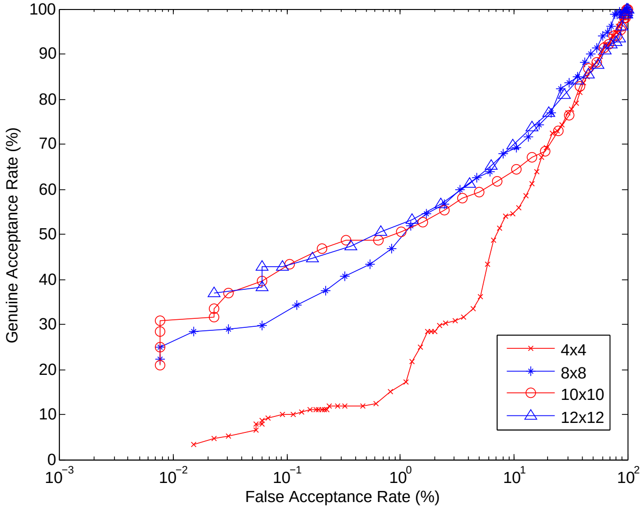


Figure 16: Algorithm 1: IITK database

CASIA: ROC curve

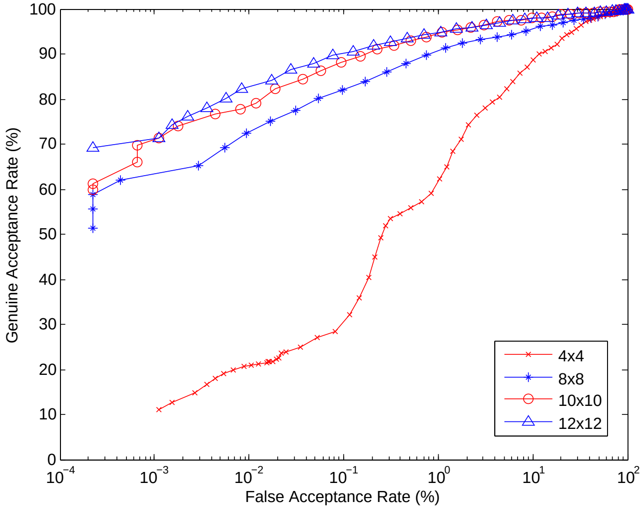


Figure 17: Algorithm 1: CASIA database

PolyU: ROC curve

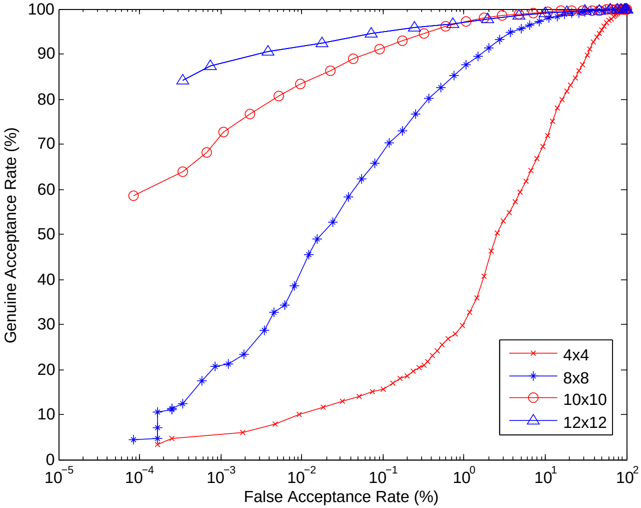


Figure 18: Algorithm 1: PolyU database

IITK

IITK				
	$k = 5$	$k = 11$	$k = 15$	$k = 21$
Accuracy (%)	100	100	100	100
FAR (%)	0	0	0	0
FRR (%)	0	0	0	0
EER (%)	0	0	0	0

Table 4: Performance of proposed system: Algorithm 2 on IITK database

CASIA

CASIA				
	$k = 5$	$k = 11$	$k = 15$	$k = 21$
Accuracy (%)	**99.89**	99.88	99.87	99.88
FAR (%)	**0.000298**	0.0029	0.037	0.00021
FRR (%)	**0.2157**	0.2364	0.2157	0.2364
EER (%)	**0.2173**	0.240	0.217388	0.238

Table 5: Performance of proposed system: Algorithm 2 on CASIA database

PolyU

PolyU				
	$k = 5$	$k = 11$	$k = 15$	$k = 21$
Accuracy (%)	**99.9673**	99.9673	99.9673	99.9673
FAR (%)	**0.0006729**	0.000673	0.00067	0.0006729
FRR (%)	**0.064766**	0.0648	0.0648	0.0648
EER (%)	**0.0648**	0.0651874	0.06476684	0.064851

Table 6: Performance of proposed system: Algorithm 2 on PolyU database

IITK: ROC curve

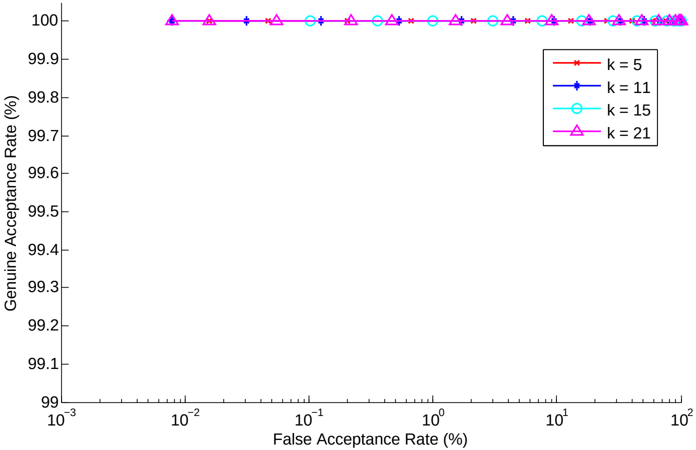


Figure 19: Algorithm 2: IITK database

CASIA: ROC curve

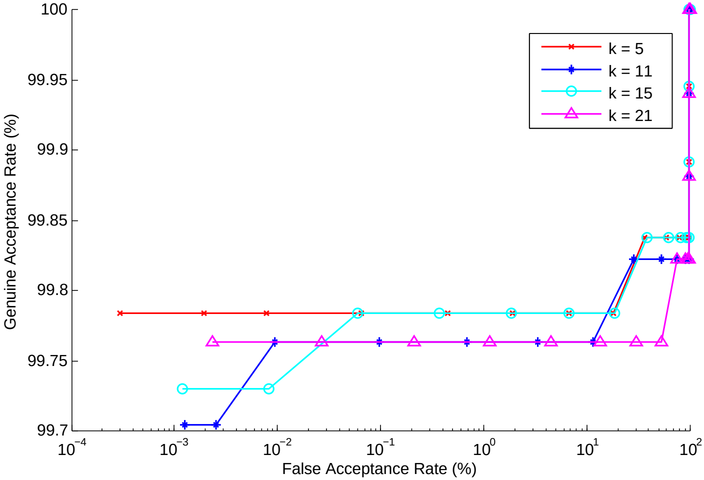


Figure 20: Algorithm 2: CASIA database

PolyU: ROC curve

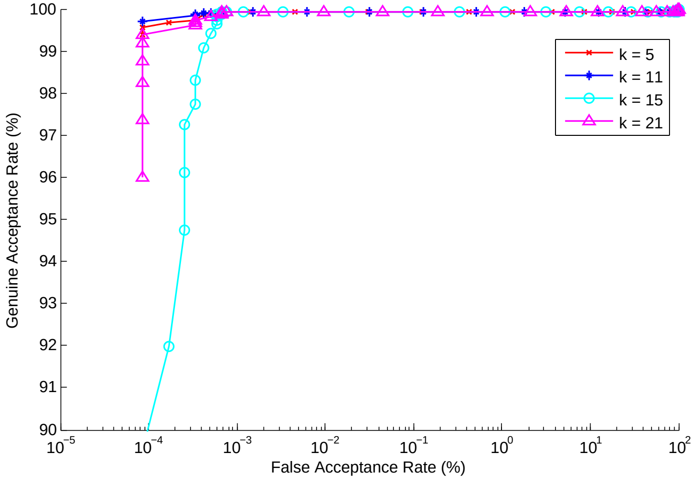


Figure 21: Algorithm 2: PolyU database

Verification: Algorithm 1			
No. of Sub-images	PolyU	CASIA	IITK
4×4	18.31 %	10.74 %	25.68 %
8×8	4.77 %	5.69 %	22.36 %
10×10	1.99 %	3.30 %	22.36 %
12×12	1.88 %	3.49 %	22.41 %
16×16	2.08 %	7.24 %	22.85 %

Table 7: EERs of the proposed system: Algorithm 1

Verification: Algorithm 2			
kernel (k)	PolyU	CASIA	IITK
5	0.0648 %	0.2173 %	0 %
11	0.0651 %	0.240 %	0 %
15	0.0648 %	0.2173 %	0 %
21	0.0648 %	0.238 %	0 %

Table 8: EERs of the proposed system: Algorithm 2

Identification

A subject image from testing set is picked up and queried against the entire training set. The best match is obtained. If the best match corresponds to the subject queried then such a match is said to be a *correct match*. Similarly, N subjects from testing set are queried. The identification based accuracy of the system is obtained as

$$\text{Accuracy} = \frac{\sum_{i=1}^{N} q_i}{N} \times 100\% \tag{25}$$

where q_i is defined as

$$q_i = \begin{cases} 1 & \text{if } i^{th} \text{ subject is correctly identified by the system} \\ 0 & \text{otherwise} \end{cases} \tag{26}$$

Top 1 match Accuracy: Algorithm 1			
No. of Sub-images	PolyU	CASIA	IITK
4×4	88.76 %	97.94 %	91.44 %
8×8	99.96 %	99.06 %	93.42 %
10×10	99.74 %	99.43 %	92.10 %
12×12	99.77 %	99.44 %	92.11 %
16×16	99.87 %	98.00 %	89.47 %

Table 9: Identification Accuracy of the proposed system: Top best match

Top 1 match Accuracy: Algorithm 2			
kernel (k)	PolyU	CASIA	IITK
5	100 %	100 %	100 %
11	100 %	100 %	100 %
15	100 %	100 %	100 %
21	100 %	100 %	100 %

Table 10: Identification Accuracy of the proposed system: Top best match

Comparison of Algorithm 1 and Algorithm 2 on IITK database

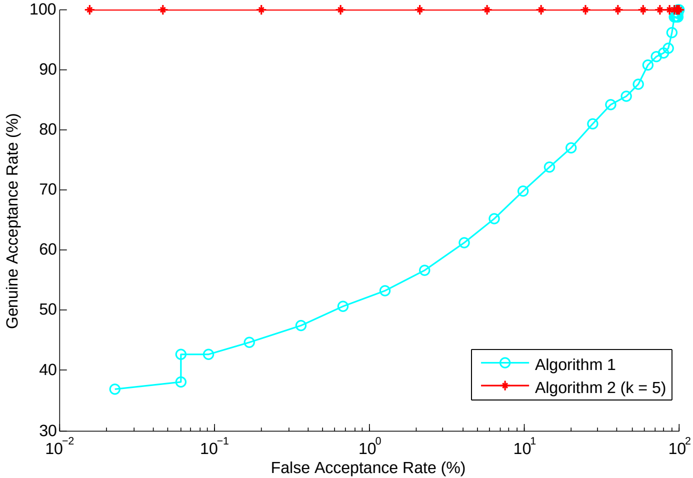


Figure 22: IITK database

Comparison of Algorithm 1 and Algorithm 2 on CASIA database

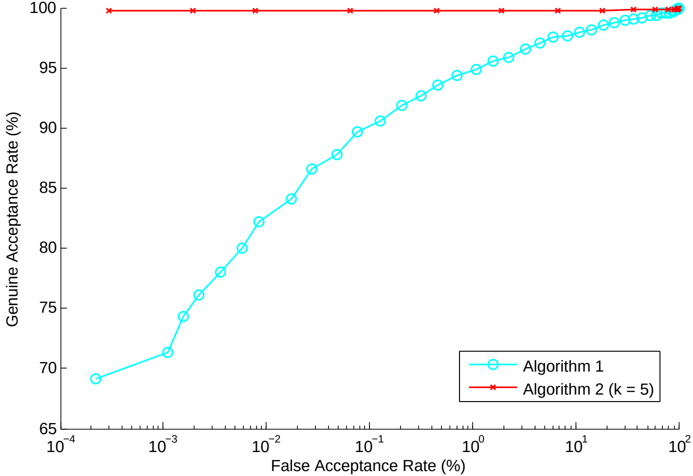


Figure 23: CASIA database

Comparison of Algorithm 1 and Algorithm 2 on PolyU database

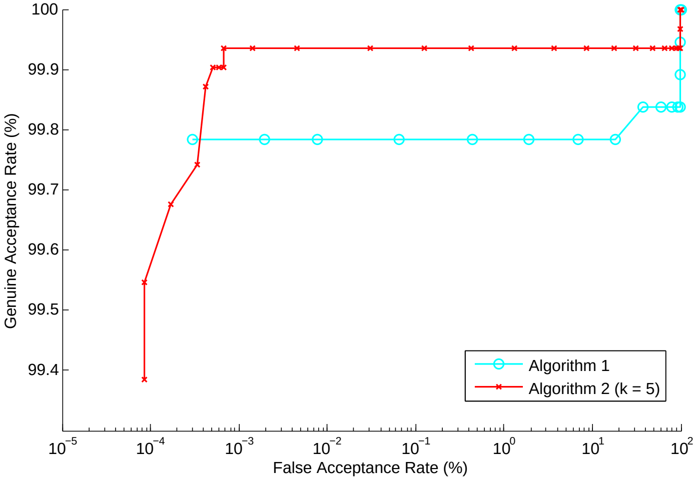


Figure 24: PolyU database

Comparison of proposed system with [8] and [9]

	Proposed system: Algorithm 1	Proposed system: Algorithm 2	Zhang [8]	Naresh [9]
ACC (%)	98.38	**99.96**	98.56	99.31
FAR (%)	0.80	**0.0006**	0.44	0.28
FRR (%)	2.42	**0.0648**	4.80	1.93
EER (%)	1.88	**0.0647**	3.76	0.91

Table 11: Comparison of proposed system with [8] and [9]

Robustness to occlusion

- The performance of proposed method is evaluated on occluded images
- The images are $0.1W \times 0.1H$, $0.2W \times 0.2H$, $0.3W \times 0.3H$, $0.4W \times 0.4H$, $0.5W \times 0.5H$ occluded

(a) (b) (c) (d) (e)

- The experiment is performed on PolyU database and the palmprint image is divided into 10×10 sub-images
- Occluded sub-images do not participate in matching

Robustness to occlusion

PolyU					
	$0.1W \times 0.1H$	0.2×0.2	0.3×0.3	0.4×0.4	0.5×0.5
ACC (%)	98.02	97.95	97.80	97.25	96.34
FAR (%)	0.99	1.27	2.28	3.11	2.90
FRR (%)	2.95	2.82	2.14	2.39	4.40
EER (%)	2.78	3.42	2.28	3.11	4.46

Table 12: Robustness to occlusion

Robustness to occlusion

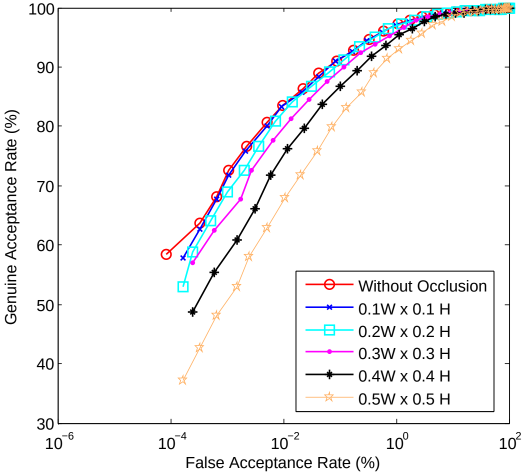


Figure 25: Robustness to occlusion

Section

Conclusions

- Features extracted using Local Structure Tensor (LST) can be effectively used for person recognition using palmprint
- The idea of extracting features from sub-images is to tackle the problem of occlusion
- The force field based filter strengthens the feature extraction ability of LST
- Using the orientation from Force Field Filter, LST matrix can be evaluated for only those specific pixels of the sub-image which belong to the dominant orientation bin
- This is crucial because it reduces computational time in the feature extraction step

Contd..

The Proposed System

- is robust to partial occlusion. Hence, a person with injuries can also be recognized by it
- performs well with low resolution palmprint images. Hence, even a low cost scanner can be used to achieve efficient performances
- performs equally well with acquisition devices which have constraints using pegs as well as those which are pegs-free. Hence, physically challenged people can also be enrolled and recognized

Future Work

- The approach of force field based filter has been tested with only LST in the feature extraction step
- The approach is generic enough and can be applied to other feature extraction techniques as well
- The following aspects can be considered for further study
    - this work has used only the orientation part of the force field. But it is worth to see the effect of the magnitude of the force
    - In Algorithm 1, the features are obtained from a mean matrix obtained from LST matrices of each individual pixels falling in the sub-image region. It is to be seen the possibility of finding the contribution of a pixel to the sub-image (for example, contribution of a pixel to texture in case of palmprint) so that this contribution can be used as a weight of a pixel, instead of using mean. This may improve the performance

Section

References I

[1] D. J. Hurley, M. S. Nixon, and J. N. Carter, "Force field energy functionals for image feature extraction," *Image and Vision Computing*, vol. 20, no. 5-6, pp. 311–317, 2002.

[2] D. Zhang, *Palmprint Authentication*. Kluwer Academic Publishers, 2004.

[3] H. Knutsson, "A tensor representation of 3-D structures," in *5th IEEE-ASSP and EURASIP Workshop on Multidimensional Signal Processing, Noordwijkerhout, The Netherlands*, 1987.

[4] J. Bigun, G. Granlund, and J. Wiklund, "Multidimensional orientation estimation with applications to texture analysis and optical flow," *IEEE-PAMI*, vol. 13, no. 8, pp. 775–790, 1991.

References II

[5] N. K. Kachhi, “An Efficient Occlusion Invariant Palmprint based Verification System,” Master’s thesis, Indian Institute of Technology Kanpur, 2009.

[6] L. Ma, T. Tan, Y. Wang, and D. Zhang, “Local intensity variation analysis for iris recognition,” *Pattern Recognition*, vol. 37, no. 6, pp. 1287–1298, 2004.

[7] R. W. R.C. Gonzalez and S. Eddins, *Digital Image Processing Using MATLAB(R)*.
Prentice Hall, 2004.

[8] D. Zhang, W.-K. Kong, J. You, and M. Wong, “Online palmprint identification,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 9, pp. 1041–1050, 2003.

References III

[9] N. K. P. G. G. S. Badrinath, "Palmprint based verification system robust to occlusion using low-order zernike moments of sub-image," in *British Machine Vision Conference*, BMVC 2009.

Thank You

Questions? and Suggestions