Tri-Modal Biometric Fusion for Human Authentication by Tracking Differential Code Pattern

Aditya Nigam
School of Computing and Electrical Engineering (SCEE)
Indian Institute of Technology, Mandi
(IIT-Mandi)

Indian Institute of Technology Mandi

NCVPRIPG-2015 Presentation at IIT Patna

December 19, 2015

Problem Definition

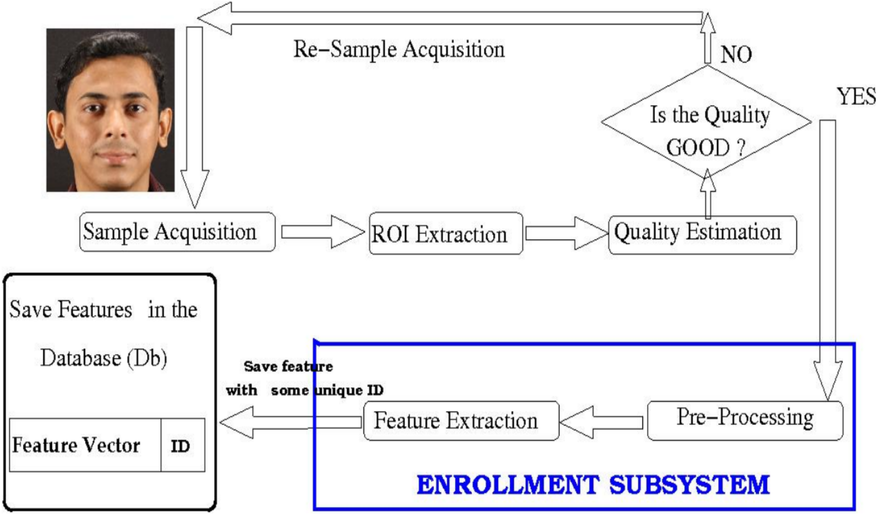
- Biometric based Personal authentication systems are in demand.
- Several biometric traits are studied such as face, iris, palmprint, ear, fingerprint etc.
- Biometrics based PAS:

  Enrollment Problem: Features extracted are saved in database to enroll any Subject.

  Authentication Problem: One to One matching and decide using thresholding (Verification).

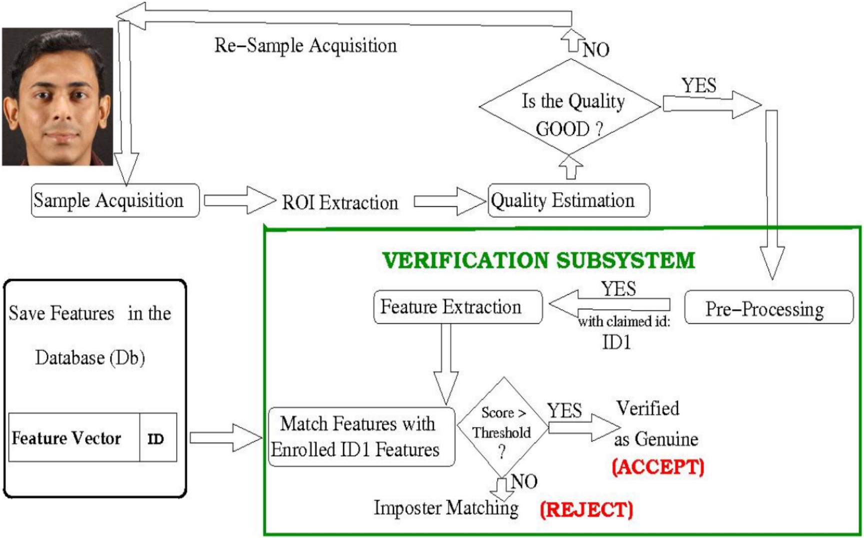
  Identification Problem: One to Many matching and best matching scores and corresponding subjects are reported (Recognition problem)

Enrollment



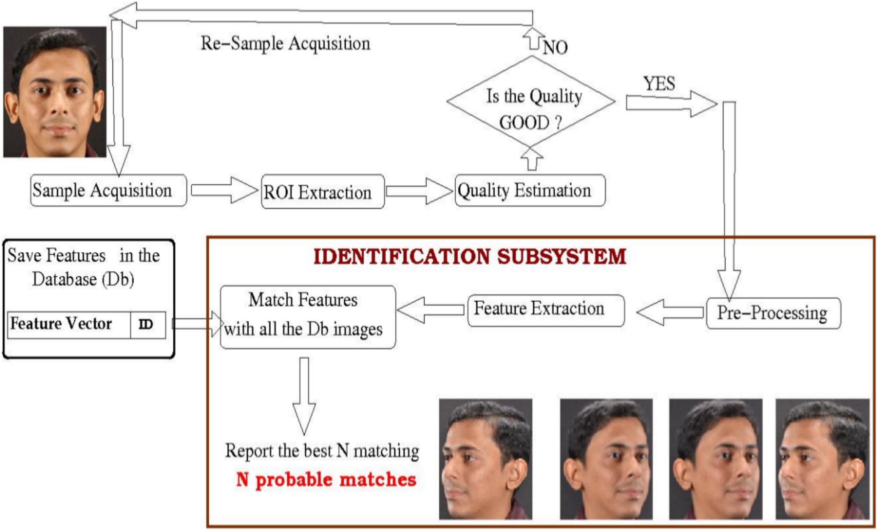
Flow diagram of Enrollment Process

Authentication - (EER)



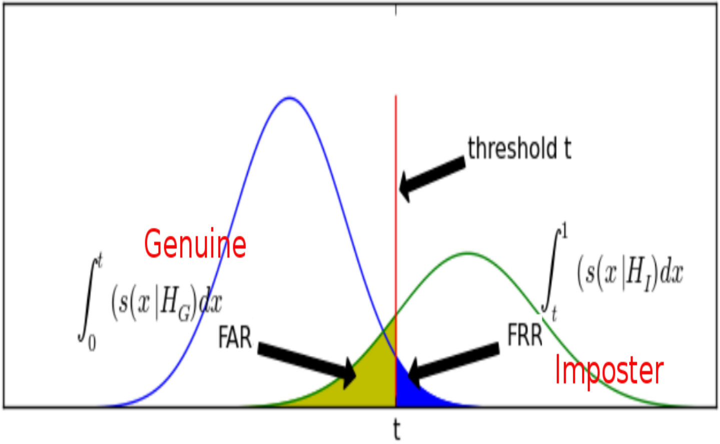
Flow diagram of Authentication Process

Identification - (CRR)



Flow diagram of Identification Process

Genuine Vs Imposter Graph



Several Biometric Traits Available

(a) Face

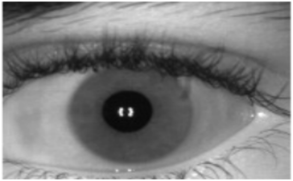
(b) Fingerprint

(c) 4-Slap

(d) Ear

(e) Vein Pattern

(f) Footprint



(g) Iris

(h) Knuckleprint

(i) Palmprint

Biometric Trait's Properties

1. **Uniqueness:** The features associated with the biometric trait should be different for everyone.
2. **Universality:** The biometric trait should be owned by everyone and should not be lost easily.
3. **Circumvention:** The biometric trait should not be spoofed or forged easily.
4. **Collectability:** The biometric trait should be able to acquire by some digital sensor.
5. **Permanence:** The features associated with the biometric trait should be time invariant (*i.e.* temporally stable).
6. **Acceptability:** The biometric trait should be accepted by the society without any objection.

Trait-wise Challenges and Issues

Modality	Motivation	Challenges	Issues
Face [58]	Most obvious, Non intrusive, Acceptance, Universality, Cheap Sensors	Pose, Expression, Illumination, Aging, Rotation, Translation, Occlusion and Background	Too many Challenges
Fingerprint [16]	Unique, Easier to acquire, Less cooperative, Cheap Sensor	Rotation and Translation	Acceptance
Iris [72]	Unique, Well Protected, Highly Discriminative	Segmentation, Occlusion and Illumination, Rotation, Off-angle, Motion Blur	Acquisition, Cooperation, Acceptance
Palmprint [11]	Touch-less, Lesser Intrusive, Bigger ROI, Faster	Illumination, Rotation and Translation	Acquisition, Cooperation, Acceptance
Knuckleprint [84]	Unique, Well Protected, Highly Discriminative, Robust Features, Cheap Sensor	Illumination, Rotation and Translation	Acquisition, Cooperation, Acceptance
Ear [42]	Non-Intrusive, Acceptance, Universal, Cheap Sensor, Robust Shape	Illumination, Rotation in/out plane, Scale, Pose and Translation	Acquisition, Cooperation, Acceptance

Motivation (Multimodal)

Why do we require several biometric traits

Because none of them can be considered as perfect.

- The performance of any unimodal biometric system is often got restricted by varying environmental and uncontrolled conditions.
- Also the performance got restricted by sensor precision and reliability as well as several trait specific challenges such as pose, expression, aging etc for face recognition.
- Hence fusing more than one biometric samples, traits or algorithms in pursuit of superior performance can be very useful idea.

Motivation (Traits)

- Out of the all the traits listed in previous slide fingerprint is used and accepted widely worldwide. But stills cons are Fail to acquire specially for cultivators and workers, low public acceptance as connected to criminals and Dirty.

Motivation (Traits)

- Out of the all the traits listed in previous slide fingerprint is used and accepted widely worldwide. But stills cons are Fail to acquire specially for cultivators and workers, low public acceptance as connected to criminals and Dirty.

Pros of Iris, Knuckle and Palmprint

- Iris have the most unique and discriminative texture (Considered as the best biometric trait).
- No expression and pose (All three).
- No occlusion, less cooperation an inexpensive sensors (Knuckle and Palmprint).
- Larger ROI ensures abundance of structural features including principle lines, wrinkles, creases and texture pattern even in low resolution palmprint images (Knuckle and Palmprint).

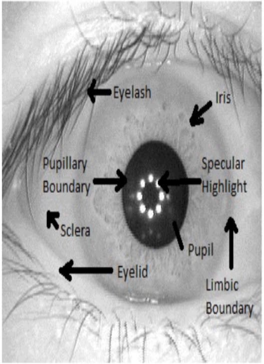
Fingerprint Vs Knuckleprint

(a) fp 1 (b) fp 2 (c) fp 3 (d) fp 4 (e) fp 5 (f) fp 6 (g) fp 7

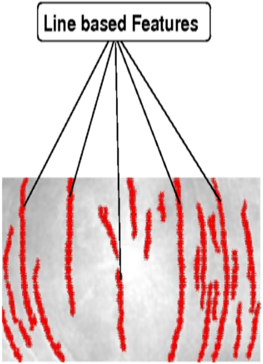
(h) fkp 1 (i) fkp 2 (j) fkp 3 (k) fkp 4 (l) fkp 5 (m) fkp 6 (n) fkp 7

Fingerprint Vs Knuckleprint (Row 1 shows fingerprints while second row shows the corresponding knuckleprints)

Discriminative Features



(a) Iris Anatomy



(b) Knuckle Anatomy

(c) Palmprint Anatomy

Stages in any Biometric System

Stages in any Biometric System

1. **Sample Acquisition :** Initially the raw data is captured using the data sensor. It may require a setup and its own software routine. This is very critical and important stage as what ever is acquired in this stage will be used as the input to all the subsequent stages.

Stages in any Biometric System

1. **Sample Acquisition :** Initially the raw data is captured using the data sensor. It may require a setup and its own software routine. This is very critical and important stage as what ever is acquired in this stage will be used as the input to all the subsequent stages.
2. **ROI Extraction :** From the raw data the actual region of interest (ROI) is extracted in this stage.

1. **Sample Acquisition :** Initially the raw data is captured using the data sensor. It may require a setup and its own software routine. This is very critical and important stage as what ever is acquired in this stage will be used as the input to all the subsequent stages.
2. **ROI Extraction :** From the raw data the actual region of interest (ROI) is extracted in this stage.
3. **Quality Estimation :** The quality assessment of the sample ROI extracted is done in this stage. If the quality of the obtained ROI is poor then re-acquisition of the sample is done.

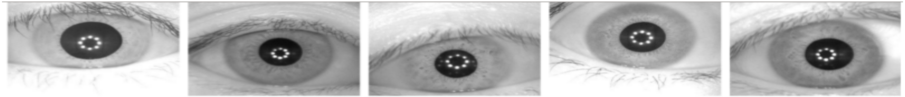
Stages in any Biometric System

1. **Sample Acquisition :** Initially the raw data is captured using the data sensor. It may require a setup and its own software routine. This is very critical and important stage as what ever is acquired in this stage will be used as the input to all the subsequent stages.
2. **ROI Extraction :** From the raw data the actual region of interest (ROI) is extracted in this stage.
3. **Quality Estimation :** The quality assessment of the sample ROI extracted is done in this stage. If the quality of the obtained ROI is poor then re-acquisition of the sample is done.
4. **Sample Preprocessing :** In this stage the obtained ROI is preprocessed using several enhancement techniques. Also some transformations can be performed over it to get robust sample image representation.

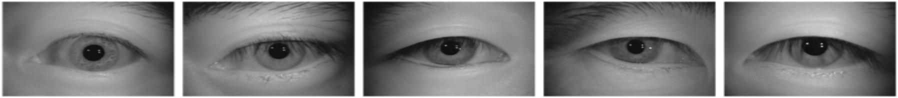
Stages in any Biometric System

1. **Sample Acquisition :** Initially the raw data is captured using the data sensor. It may require a setup and its own software routine. This is very critical and important stage as what ever is acquired in this stage will be used as the input to all the subsequent stages.
2. **ROI Extraction :** From the raw data the actual region of interest (ROI) is extracted in this stage.
3. **Quality Estimation :** The quality assessment of the sample ROI extracted is done in this stage. If the quality of the obtained ROI is poor then re-acquisition of the sample is done.
4. **Sample Preprocessing :** In this stage the obtained ROI is preprocessed using several enhancement techniques. Also some transformations can be performed over it to get robust sample image representation.
5. **Feature Extraction and Matching :** In this stage robust feature vectors are computed and stored. Then they are used to compute a score for any given matching which can decide whether a it is genuine or imposter.

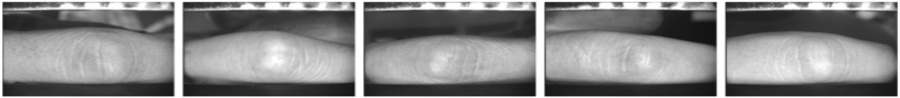
Sample Database Images



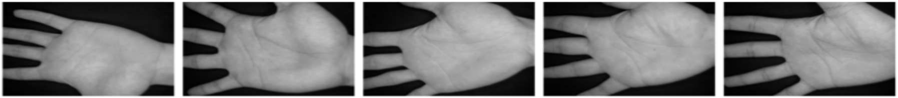
(a) Casia Interval Iris Database



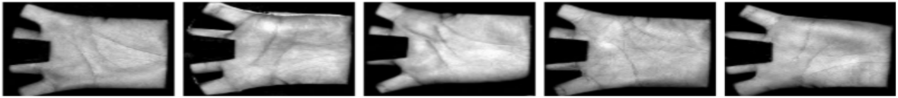
(b) Casia Lamp Iris Database



(c) PolyU Knuckleprint Database

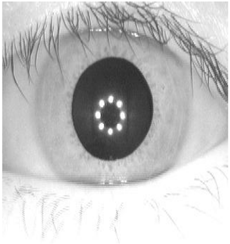


(d) Casia Palmprint Database

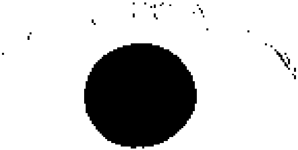


(e) PolyU Palmprint Database

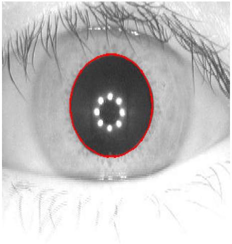
▸ Seg



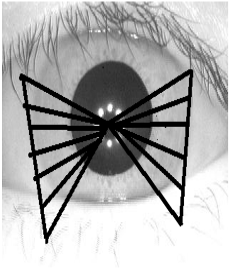
(a) Original

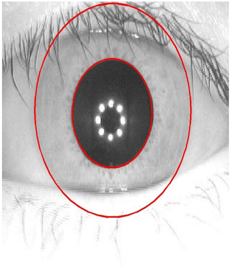


(b) Thresholded

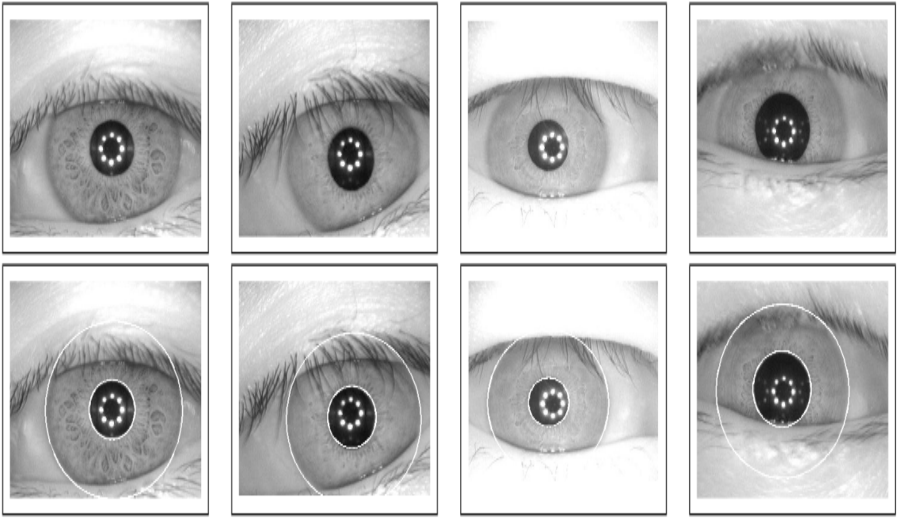


(c) Segmented Pupil



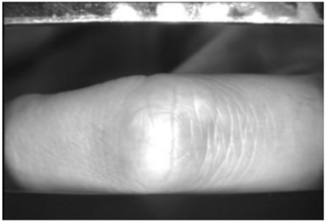


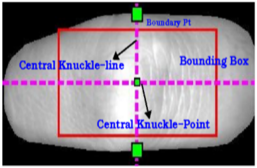
Segmented Iris



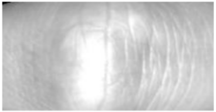
Knuckleprint ROI Extraction - FKP Area

▸ Seg


Raw knuckleprint

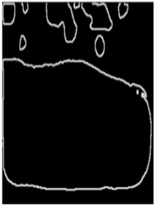


Annotated Knuckleprint

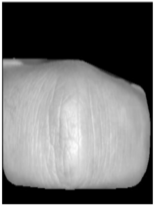

Knuckleprint ROI


(a) Raw

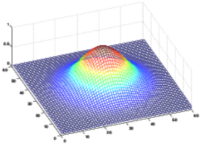

(b) Binary


(c) Canny

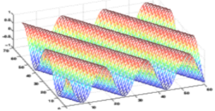

(d) Boundary


(e) FKP Area

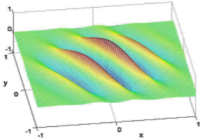
Knuckleprint ROI Extraction - Gabor Filter



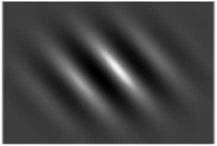
(a) 2D Gaussian



(b) 2D Sinusoid



(c) 2D Gabor



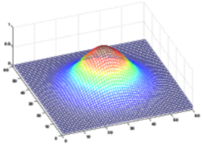
(d) 2D Gabor Filter

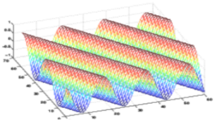
$$G(x, y; \gamma, \theta, \psi, \lambda, \sigma) = \underbrace{e^{-\left(\frac{X^2+Y^2\cdot\gamma^2}{2\cdot\sigma^2}\right)}}_{\text{Gaussian Envelope}} \times \underbrace{e^{i\left(\frac{2\pi X}{\lambda}+\psi\right)}}_{\text{Complex Sinusoid}}$$

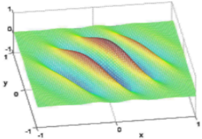
$$X = x * cos(\theta) + y * sin(\theta)$$

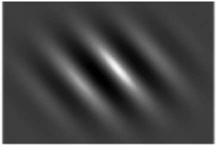
$$Y = -x * sin(\theta) + y * cos(\theta)$$

Knuckleprint ROI Extraction - Gabor Filter


(a) 2D Gaussian


(b) 2D Sinusoid


(c) 2D Gabor


(d) 2D Gabor Filter

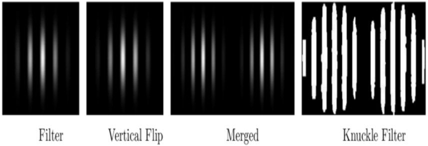
$$G(x, y; \gamma, \theta, \psi, \lambda, \sigma) = \underbrace{e^{-\left(\frac{X^2 + Y^2 \cdot \gamma^2}{2 \cdot \sigma^2}\right)}}_{\text{Gaussian Envelope}} \times \underbrace{e^{i\left(\frac{2\pi X}{\lambda} + \psi\right)}}_{\text{Complex Sinusoid}}$$

$$X = x * cos(\theta) + y * sin(\theta) \qquad X = x * cos(\theta) + y * sin(\theta) + c * (-x * sin(\theta) + y * cos(\theta))^2$$

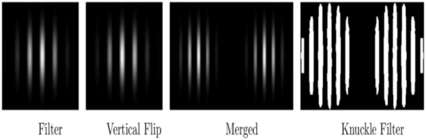
$$Y = -x * sin(\theta) + y * cos(\theta) \qquad Y = -x * sin(\theta) + y * cos(\theta)$$

c is Curvature Parameter

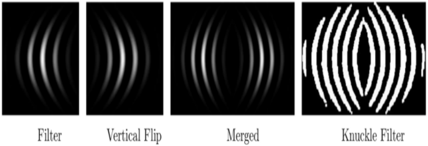
Knuckleprint ROI Extraction - Knuckle Filter



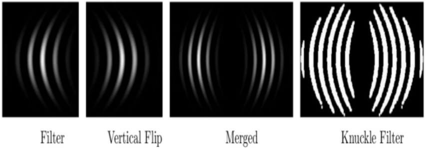
(a) c=0.00,d=0



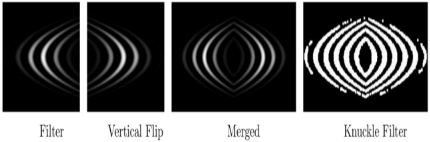
(b) c=0.00,d=30



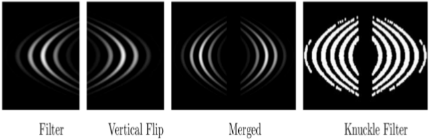
(c) c=0.01,d=0



(d) c=0.01,d=30



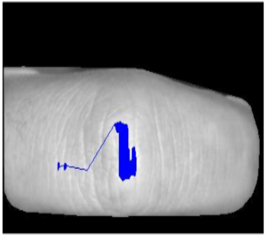
(e) c=0.04,d=0



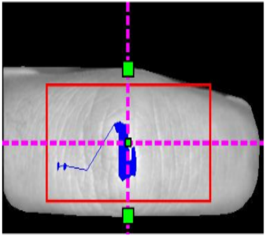
(f) c=0.04,d=30

Knuckleprint ROI Extraction

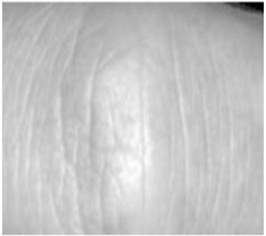
Curvature Knuckle Filter



Knuckle Filter Response

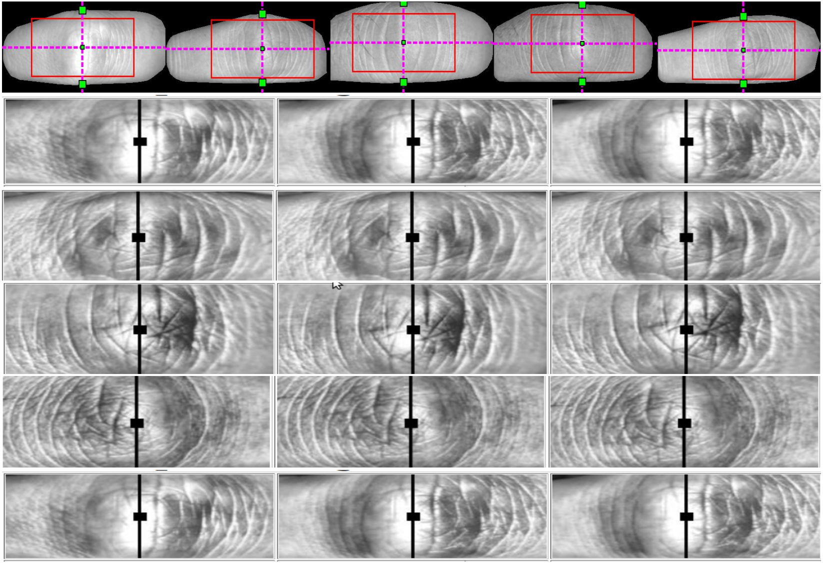


Annotated



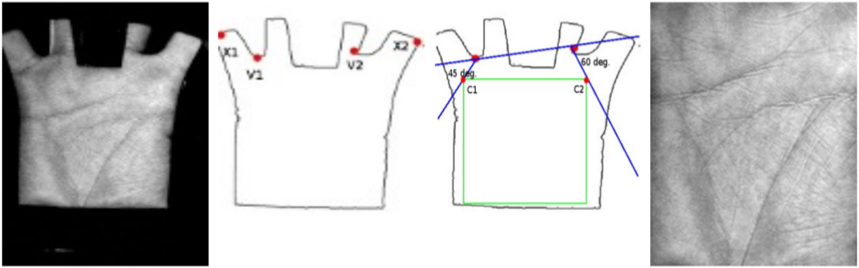
Knuckle ROI

Correctly Segmented Knuckleprint ROI



Palmprint ROI Extraction

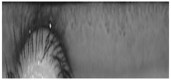
Seg



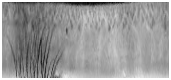
(a) Original (b) Contour (c) Key Points (d) Palmprint ROI

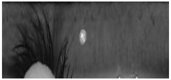
Sample Biometric ROI Images

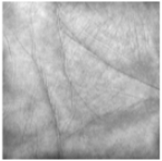

(a) Iris A

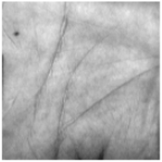

(b) Iris B

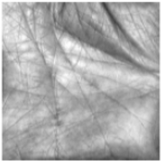

(c) Iris C

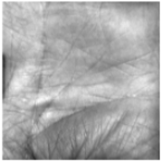

(d) Iris D

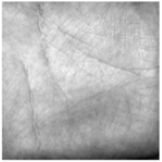

(e) Iris E

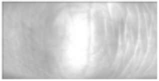

(f) Polyu A

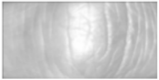

(g) Polyu B


(h) Polyu C


(i) Polyu D

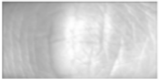

(j) Polyu E


(k) Polyu A


(l) Polyu B


(m) Polyu C

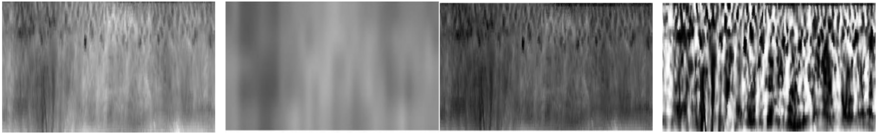

(n) Polyu D


(o) Polyu E

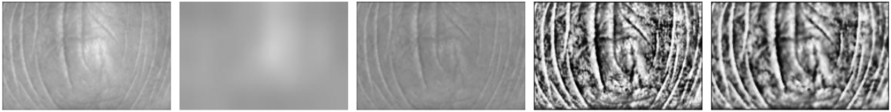
Biometric Sample ROI Enhancement

- The sample ROI is divided into blocks and the mean of each block is considered as the coarse illumination of that block which is expanded to the original block size.
- The estimated illumination of each block is subtracted from the corresponding block of the original image to obtain the uniformly illuminated ROI.
- The contrast of the resultant ROI is enhanced using Contrast Limited Adaptive Histogram Equalization ($CLAHE$).
- Finally, Wiener filter is applied to reduce noise to obtain the enhanced texture.

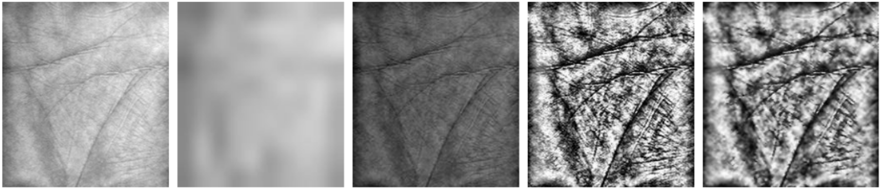
Biometric Sample ROI Enhancement



(a) Original Iris (b) Esti. Illumination (c) Uniform Illumination (d) Weiner Filtering



(a) Original (b) Bg Illum. (c) Uni. Illum. (d) Enhanced (e) Noise Removal



(a) Original (b) Bg Illum. (c) Uni. Illum. (d) Enhanced (e) Noise Removal

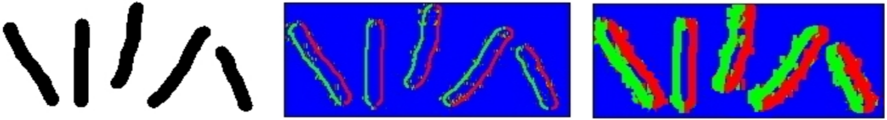
Differential Code Pattern (DCP) based Transformation

- X/Y- direction, *scharr* kernel is applied to obtain DCP^v and DCP^h. For any pixel (x, y) with $Neigh[k]$, $k = 1, 2, ...8$ as gradients of 8 neighborhood, the k^{th} bit for its 8-bit code is given by

$$dcp^{xy}[k] = \begin{cases} 1 & \textbf{if } Neigh[k] > 0 \\ 0 & \textbf{otherwise} \end{cases} \quad (1)$$

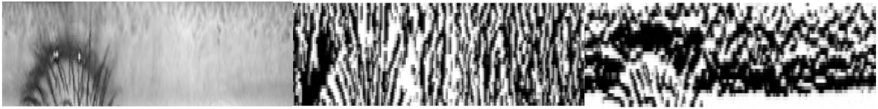
$$DCP^v(x, y) = \sum_{i=0}^{7} dcp^{xy}[i] * 2^i \quad (2)$$

- In DCP^v and DCP^h, every pixel is represented by its dcp^{xy} value, which is an encoding of edge pattern in its 8-neighborhood



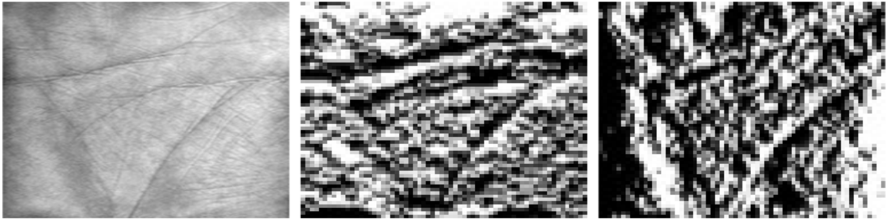
(a) Original (b) Transformed (kernel=3) (c) Transformed (kernel=9)

Differential Code Pattern (DCP) based Transformation



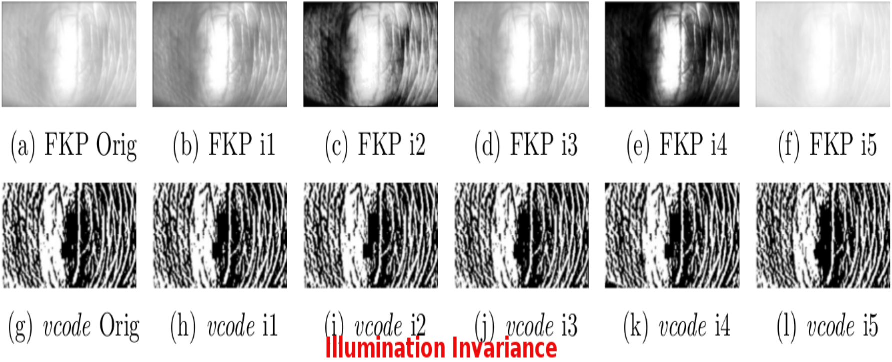
(a) Original Iris (d) Iris *vcode* (g) Iris *hcode*

(a) Orig. Knuckleprint (b) Knuckle *vcode* (c) Knuckle *hcode*

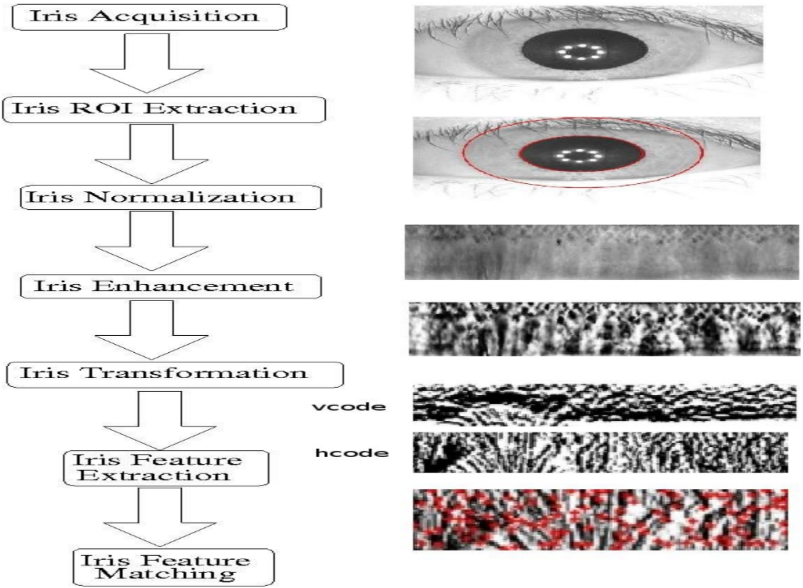


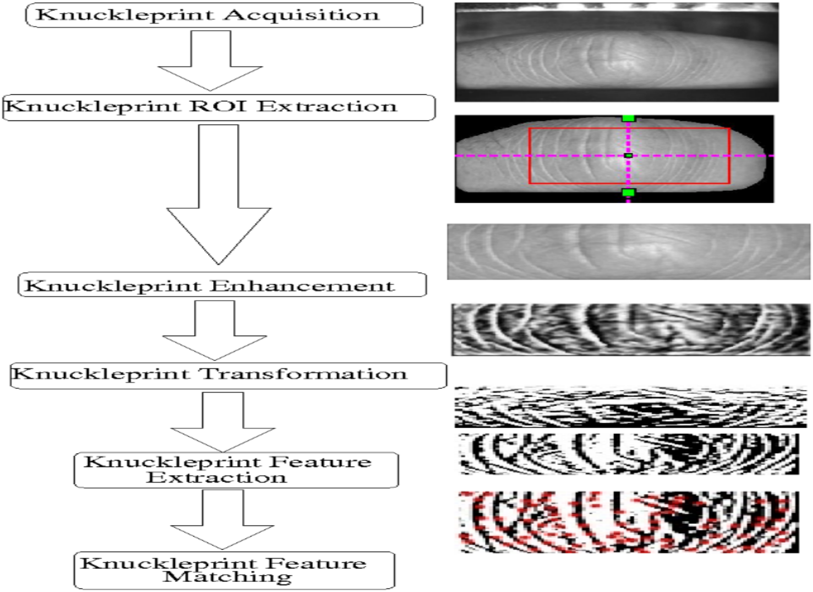
(a) Orig. Palmprint (b) Palm *vcode* (c) Palm *hcode*

Differential Code Pattern (DCP) - Illumination Invariance

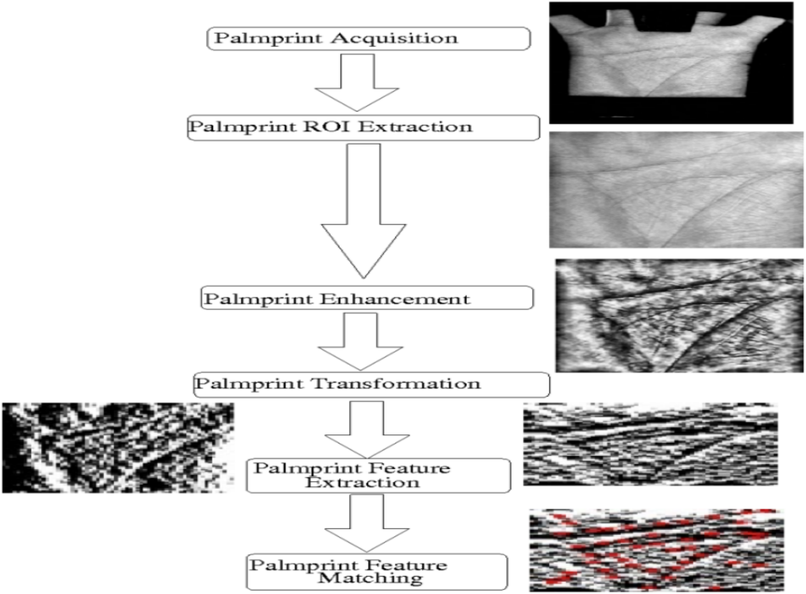


Iris Based Recognition System: Steps Involved




Knuckleprint Acquisition
Knuckleprint ROI Extraction
Knuckleprint Enhancement
Knuckleprint Transformation
Knuckleprint Feature Extraction
Knuckleprint Feature Matching

Palmprint Based Recognition System: Steps Involved



Feature Extraction [2]

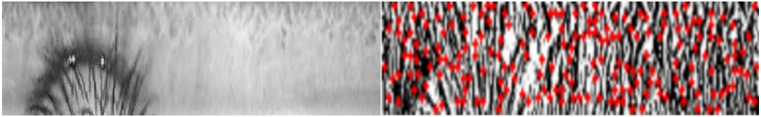
- Eigen values of autocorrelation matrix M is used to calculate good corner features.
- Matrix M can be defined for any pixel at i^{th} row and j^{th} column of *edgecode* as:

$$M(i,j) = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \tag{3}$$

such that

$$\begin{aligned}
A &= \sum_{-K \le a,b \le K} w(a,b).I_x^2(i+a,j+b) \\
B &= \sum_{-K \le a,b \le K} w(a,b).I_x(i+a,j+b).I_y(i+a,j+b) \\
C &= \sum_{-K \le a,b \le K} w(a,b).I_y(i+a,j+b).I_x(i+a,j+b) \\
D &= \sum_{-K \le a,b \le K} w(a,b).I_y^2(i+a,j+b)
\end{aligned}$$

where $w(a,b)$ is the weight given to the neighborhood, $I_x(i+a,j+b)$ and $I_y(i+a,j+b)$ are the partial derivatives sampled within the $(2K+1) \times (2K+1)$ window centered at each selected pixel.



Matching (Lukas Kanade Tracking) [1]

- **Brightness Consistency**: Features do not change much for small $\triangle t$

$$I(x, y, t) \approx I(x + \triangle x, y + \triangle y, t + \triangle t) \tag{4}$$

- **Temporal Persistence**: Features moves only within a small neighborhood for small $\triangle t$.

$$I_x V_x + I_y V_y = -I_t \tag{5}$$

where V_x, V_y are the respective components of the optical flow velocity for pixel $I(x, y, t)$ and I_x, I_y and I_t are the derivatives in the corresponding directions.

- **Spatial Coherency**: Spatial coherency assumes that a local patch of of size 5×5 neighborhood (*i.e* 25 neighboring pixels, $P_1, P_2 \dots P_{25}$) moves coherently .

$$\underbrace{\begin{pmatrix} I_x(P_1) & I_y(P_1) \\ \vdots & \vdots \\ I_x(P_{25}) & I_y(P_{25}) \end{pmatrix}}_{\text{C}} \times \underbrace{\begin{pmatrix} V_x \\ V_y \end{pmatrix}}_{\text{V}} = - \underbrace{\begin{pmatrix} I_t(P_1) \\ \vdots \\ I_t(P_{25}) \end{pmatrix}}_{\text{D}} \tag{6}$$

where rows of the matrix C represent the derivatives of image I in x, y directions and those of D are the temporal derivative at 25 neighboring pixels. The 2×1 matrix $\widehat{\mathbf{V}}$ is the estimated flow of the current feature point determined as

$$\widehat{\mathbf{V}} = (C^T C)^{-1} C^T (-D) \tag{7}$$

Proposed Geometric and Statistical Constraints

Algo

- **Vicinity Bound**: The euclidean distance between $a(i,j)$ and its estimated tracked location should be less than or equal to a pre-assigned threshold $\mathbf{Th_d}$.
- **Patch-wise Error Bound**: The tracking error T_{error} defined as pixel-wise sum of absolute difference between a local patch centered at $a(i,j)$ and that of its estimated tracked location patch should be less than or equal to a pre-assigned threshold $\mathbf{Th_e}$.
- **Correlation Bound**: The phase only correlation POC between a local patch centered at $a(i,j)$ and that of its estimated tracked location patch should be more than or equal to a pre-assigned threshold $\mathbf{Th_p}$.

However all tracked corners may not be the true matches, because of noise, local non-rigid distortions and less difference in inter class matching as compared with intra class matching. Hence a notion of consistent optical flow is introduced. The fraction of matchings FAILED to satisfy the above mentioned bounds is considered as the dissimilarity score.

Database Specifications

Subject	Pose	Total	Training	Testing	Genuine Matching	Imposter Matching
Casia V4 Interval (Iris)						
249 (395 Iris)	7	2,639	First 3	Rest	3,657	1,272,636
Casia V4 Lamp (Iris)						
411 (819 Iris)	20	16,212	First 10	Last 10	78,300	61,230,600

Subject	Pose	Total	Training	Testing	Genuine Matching	Imposter Matching
PolyU (Knuckleprint)						
165 (660 Knuckles)	12	7920	First 6	Last 6	23,760	15,657,840

Subject	Pose	Total	Training	Testing	Genuine Matching	Imposter Matching
Casia (Palmprint Left hand)						
290 Palms	8	2,320	First 4	Last 4	4,640	1,340,960
Casia (Palmprint Right hand)						
276 Palms	8	2,208	First 4	Last 4	4,416	1,214,400
Casia (Palmprint Left + Right hand)						
566 Palms	8	4,528	First 4	Last 4	9,056	5,116,640
PolyU (Palmprint Left hand)						
193 Palms	20	3,860	First 10	Last 10	19,300	3,705,600
PolyU (Palmprint Right hand)						
193 Palms	20	3,860	First 10	Last 10	19,300	3,705,600
PolyU (Palmprint Left + Right hand)						
386 Palms	20	7,720	First 10	Last 10	38,600	14,861,000

Performance Analysis - Unimodal Iris

ROC

Systems	Interval		
	DI	CRR%	EER%
Daugman [24]	1.96	99.46	1.88
Li Ma [67]	-	95.54	2.07
Masek [50]	1.99	99.58	1.09
K. Roy [67]	-	97.21	0.71
Proposed	**2.02**	**100**	**0.109**
	Lamp		
Daugman [24]	1.2420	98.90	5.59
Proposed	**1.50**	**99.87**	**1.300**

Performance Analysis - Unimodal Knuckleprint

Algorithm	Equal Error Rate
Compcode [39]	1.386
BOCV [3]	1.833
ImCompcode and MagCode [78]	1.210
MoriCode [32]	1.201
MtexCode [32]	1.816
MoriCode and MtexCode [32]	1.0481
vcode	1.5151
hcode	4.2929
$vcode + hcode$	**0.934343**

Performance Analysis - Unimodal Palmprint

Approach	Database	CRR %	EER %
PalmCode [82]	Palm (CASIA)	99.62	3.67
PalmCode [82]	Palm (PolyU)	99.92	0.53
CompCode [39]	Palm (CASIA)	99.72	2.01
CompCode [39]	Palm (PolyU)	99.96	0.31
OrdinalCode [72]	Palm (CASIA)	99.84	1.75
OrdinalCode [72]	Palm (PolyU)	100.00	0.08
Palm-Zernike [11]	Palm (CASIA)	99.75	2.00
Palm-Zernike [11]	Palm (PolyU)	100.00	0.2939
Proposed	Palm (CASIA)	100.00	0.1551
Proposed	Palm (PolyU)	99.95	0.4145

Tri-Modal Fusion Results(Iris+Knuckleprint+Palmprint)

Biometric Traits ⇒				**Iris** (I)		**Knuckle** (K)		**Palmprint** (P)		**Fusion**	
Db Name	Testing Strategy	Total Genuine	Total Imposter	FA EER (%)	FR CRR (%)	FA EER (%)	FR CRR (%)	FA EER (%)	FR CRR (%)	FA EER (%)	FR CRR (%)
db1[I=Interval; K=PolyU; P=Casia]	1**Tr**-1**Test**	349	121452	240	0.66	1849	5	235.33	0.66	0	0
				0.19	99.80	1.47	97.80	0.19	99.90	0	100
db1 [I=Interval; K=PolyU; P=Casia]	3**Tr**-4**Test**	3657	1276293	1392	4	21923	63	3007	9	0	0
				0.10	100	1.72	99.67	0.24	99.91	0	100
db2 [I=Interval; K=PolyU; P=PolyU]	1**Tr**-1**Test**	349	121452	240	0.66	1849	5	617.66	1.66	0	0
				0.19	99.80	1.47	97.80	0.49	99.33	0	100
db2 [I=Interval; K=PolyU; P=PolyU]	3**Tr**-4**Test**	3657	1276293	1392	4	21923	63	4872	14	0	0
				0.10	100	1.72	99.67	0.38	99.91	0	100
db3 [I=Lamp; K=PolyU; P=Casia]	1**Tr**-1**Test**	566	319790	4970.33	8.66	6027.66	10.66	385.33	0.66	0	0
				1.54	97.29	1.88	97.17	0.11	99.94	0	100
db3 [I=Lamp; K=PolyU; P=Casia]	4**Tr**-4**Test**	9056	5116640	66217	117	99902	177	7912	14	0	0
				1.29	99.77	1.95	99.55	0.15	100	0	100
db4 [I=Lamp; K=PolyU; P=PolyU]	1**Tr**-1**Test**	386	148610	2435	6.33	2181	5.66	1026.6	2.66	0	0
				1.64	97.23	1.46	98.01	0.69	99.13	0	100
db4 [I=Lamp; K=PolyU; P=PolyU]	6**Tr**-6**Test**	13896	5349960	70833	184	87401	227	43504	113	385	1
				1.32	99.87	1.63	99.87	0.81	99.91	0.0071	100

TABLE I. PERFORMANCE PARAMETERS FOR UNI-MODEL AND CHIMERIC MULTIMODAL DATABASES (*db*1,*db*2,*db*3 & *db*4).

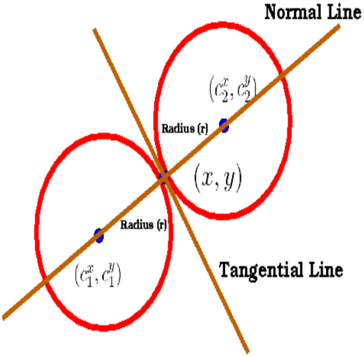
Thank You

B. D. Lucas and T. Kanade.
An Iterative Image Registration Technique with an Application to Stereo Vision.
In *IJCAI*, pages 674–679, 1981.

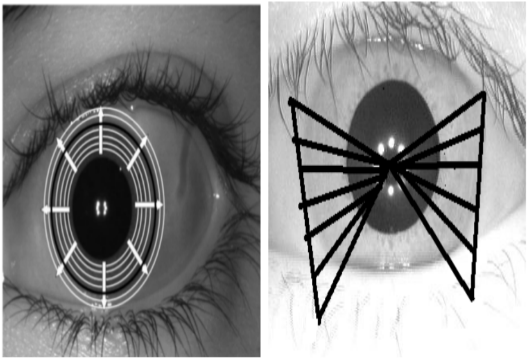
J Shi and Tomasi.
Good features to track.
In *Computer Vision and Pattern Recognition*, pages 593–600, 1994.

Iris ROI Extraction - (I)

Back



(a) Modified Hough



(b) Integro-Differential

Iris ROI Extraction - 1

Back

Algorithm 4.1 Pupil Segmentation

Require:
Iris image I of dimension $m \times n$,
$p_{r_{min}}$: minimum pupil radius,
$p_{r_{max}}$: maximum pupil radius,
t: binary threshold.

Ensure:
Pupil center c_p with co-ordinates as (c_p^x, c_p^y),
Pupil radius p_r.

$I_t \leftarrow threshold(I, t)$; // generate binary image after thresholding
$I_{tf} \leftarrow remove\_specular\_reflection(I_t)$; // flood filling
$I_g, I_{gh}, I_{gv} \leftarrow Sobel(I_{tf})$; // Sobel edge detection
$I_{gb} \leftarrow Threshold\_Gradient(I_g)$; // choosing the best edge points
$E \leftarrow$ white pixels in I_{gb}; // collect the edge points in I_{gb}
for all Edge pixels $(x, y) \epsilon E$ **do**
  $\theta(x, y) \leftarrow tan^{-1}\left(\frac{I_{gv}(x,y)}{I_{gh}(x,y)}\right)$; // edge orientation at a point
end for
$A(m, n, p_{r_{max}}) \leftarrow 0$; // 3-D array initialization for voting
for all Edge pixels $(x, y) \epsilon E$ **do**
  for $r = p_{r_{min}}$ **to** $p_{r_{max}}$ **do**
    Compute (c_1^x, c_1^y), (c_2^x, c_2^y) by putting (x, y) and $\theta(x, y)$ in eqs. (4.3)-(4.6);
    if Point (c_1^x, c_1^y) lies within I_{gb} image **then**
      $A(c_1^x, c_1^y, r) \leftarrow A(c_1^x, c_1^y, r) + 1$; // Vote Casting
    end if
    if Point (c_2^x, c_2^y) lies within I_{gb} image **then**
      $A(c_2^x, c_2^y, r) \leftarrow A(c_2^x, c_2^y, r) + 1$; // Vote Casting
    end if
  end for
end for
$c_p^x \leftarrow argmax_{(i)}\ A(i, j, k)$;
$c_p^y \leftarrow argmax_{(j)}\ A(i, j, k)$;
$p_r \leftarrow argmax_{(k)}\ A(i, j, k)$;

Iris ROI Extraction - 2 (I)

Back

Algorithm 4.2 Iris Segmentation

Require:

Iris image I of dimension $m \times n$,
$i_{r_{min}}$: minimum iris radius,
$i_{r_{max}}$: maximum iris radius,
(c_p^x, c_p^y): Pupil center,
p_r: pupil radius,
W: search window,
α_{range}: angular range defining the occlusion free sectors.

Ensure:

Iris center $c_i(c_i^x, c_i^y)$,
iris radius i_r.

$I_s \leftarrow GaussSmooth(I, \sigma = 0.5, k = 3)$; // k: kernel size, Gaussian noise removal
$max_{diff} \leftarrow 0$; // the maximum change in contour summation
for all points $(c^x, c^y) \in [W \times W]$ window around (c_p^x, c_p^y) **do**
    $prev_{sum} \leftarrow 0$; // previous circular summation of intensity values
    $start_{flag} \leftarrow True$; // no circle has yet been summed up
    for $r = i_{r_{min}}$ **to** $i_{r_{max}}$ **do**
        $c_{sum} \leftarrow 0$;
        for all $\alpha \in \alpha_{range}$ **do**
            $c_{sum} \leftarrow c_{sum} + I(c^x - r\sin(\alpha), c^y + r\cos(\alpha))$;//sector-wise summation
        end for
        $diff_{sum} \leftarrow c_{sum} - prev_{sum}$; // calculation of difference of sum
        $prev_{sum} \leftarrow c_{sum}$;
        if $diff_{sum} > max_{diff}$ and $start_{flag} \neq True$ **then**
            $max_{diff} \leftarrow diff_{sum}$;
            $c_i^x \leftarrow c^x$, $c_i^y \leftarrow c^y$, $i_r \leftarrow r$; // update the parameters
        end if
        $start_{flag} \leftarrow False$; // a circle has been summed up
    end for
end for

Back

Algorithm 6.1 Palmprint ROI Extraction

Require:

Full acquired palmprint image I of dimension $m \times n$ as shown in Fig. 6.2(a).

Ensure:

The cropped palmprint ROI as shown in Fig. 6.2(d).

1: Threshold the palmprint image I_p to extract the hand contour C.
2: Over the hand contour C find the coordinates of four key points X_1, X_2, V_1, V_2 as shown in Fig.6.2(b).
3: Compute C_1 as the intersection point of hand contour and line passing from V_1 with a slope of 45°.
4: Compute C_2 as the intersection point of hand contour and line passing from V_2 with a slope of 60°.
5: Midpoints of the line segments V_1C_1 and V_2C_2 are considered as one side of the required square palmprint ROI.
6: Extract the required square palmprint ROI as shown in Fig. 6.2(d).

Knuckleprint ROI Extraction - 1 (I)

Back

Algorithm 5.1 Knuckleprint ROI Detection

Require:
Raw Knuckleprint image I of size $m \times n$.

Ensure:
The knuckleprint ROI FKP_{ROI}, of size $(2*w+1) \times (2*h+1)$.

1: Enhance the FKP image I to I_e using CLAHE;
2: Binarize I_e to I_b using Otsu thresholding;
3: Apply Canny edge detection over I_b to get I_{cedges};
4: Extract the largest connected component in I_{cedges} as FKP raw boundary, (FKP^{raw}_{Bound});
5: Erode the detected boundary FKP^{raw}_{Bound} to obtain continuous and smooth FKP boundary, FKP^{smooth}_{Bound};
6: Extract the knuckle area K_a = All pixels in image $I \in$ the $ConvexHull(FKP^{smooth}_{Bound})$;
7: Apply the knuckle filter $F^{0.01,30}_{kp}$ over all pixels $\in K_a$;
8: Binarize the filter response using $f * max$ as the threshold;
9: The central knuckle line (c_{kl}), is assigned as that column which is having the maximum knuckle filter response;
10: The mid-point of top and bottom boundary points over $c_{kl} \in K_a$, is defined as the central knuckle point (c_{kp}).
11: The knuckle ROI (FKP_{ROI}) is extracted as the region of size $(2*w+1)\times(2*h+1)$ from raw knuckleprint image I, considering c_{kp} as its center point.

Iris Quality - 1

Back

Algorithm 4.3 Eyelid Detection

Require: Normalized Iris image NI of dimension $m \times n$, $s : (s_x, s_y)$: initial seed point in NI, t: region-growing threshold

Ensure: Eyelid region LID, Eyelid Mask $Mask_{eyelid}$

$S \leftarrow s$ // Seed point is added to eyelid set S
$M = NI(s_x, s_y)$ // Mean of set S
while True **do**
  $min \leftarrow Infinity$, $minPoint \leftarrow \phi$
  for all unallocated neighboring pixels (x, y) of S **do**
    if $|NI(x, y) - M| < min$ **then**
      $min \leftarrow |NI(x, y) - M|$ // absolute difference with region's mean
      $minPoint \leftarrow (x, y)$ // keep track of the best point
    end if
  end for
  if $min > t$ or size(S)==size(NI) **then**
    break // stop region-growing
  end if
  $S \leftarrow S \cup minPoint$ // add the best point to the set
  $M \leftarrow mean(S)$ // update the mean value
end while
$LID \leftarrow S$
$Mask_{eyelid} \leftarrow NI(LID)$ // eyelid mask

Iris Quality - 2 (I)

Back

Algorithm 4.4 Eyelid Detection

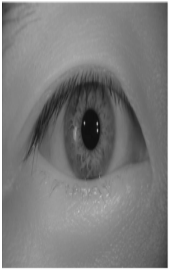
Require: Normalized Iris image NI of dimension $r \times c$

Ensure: Eyelash Mask $Maskeyelash$

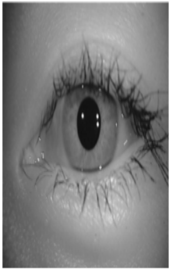
1: $SD \leftarrow filter2D(NI, std(3,3))$// 2D filtering with 3x3 standard deviation filter
2: $SD \leftarrow \frac{SD}{max(SD)}$ // normalize w.r.t. maximum value
3: $N = \frac{NI}{255}$ // Normalized intensity values(0-1) of NI
4: $F = 0.5 \times SD + 0.5 \times (1 - N)$ // fusion of std. deviation and intensity values.
5: $F_H \leftarrow imhist(F)$ // image histogram
6: $Thresh \leftarrow Otsu(F_H)$ // determine Otsu threshold
7: $Mask_{eyelash} \leftarrow threshold(F_H, Thresh)$ // Otsu thresholding: $Mask_{eyelash}(x,y)$ is set only if (x,y) is an eyelash pixel

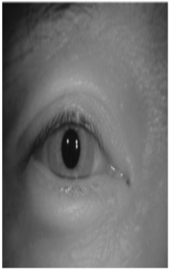
Iris Quality - 3 (I)

Back


(a) Image 1


(b) Image 2


(c) Image 3


(d) Image 4


(e) Image 5

Image	Focus	Blur	Occlusion	Contrast	Dilation	Reflection	Quality
Image 1	0.2034	0.6783	0.8834	0.8873	0.8348	0.9865	5
Image 2	0.1985	0.6138	0.7289	0.8915	0.8517	0.9937	4
Image 3	0.1536	0.4974	0.7049	0.7760	0.7739	0.9103	3
Image 4	0.1648	0.5088	0.6790	0.7954	0.7856	1.0000	2
Image 5	0.1156	0.4067	0.2819	0.6659	0.7840	0.9589	1

Knuckleprint Quality - 1 (I)

Back

Algorithm 5.2 Uniformity based Quality Attribute (S)

Require: The vle and wf pixel set for the input image (I) of size $m \times n$.

Ensure: Return the value S for the input image (I).

1. $F_{map} = and(wf, vle)$; $[focus\ mask]$
2. M_1, M_2=Mid-point of Left half $(\frac{n}{2}, \frac{n}{2})$ and Right half $(\frac{m+n}{2}, \frac{n}{2})$ of the input image (I);
3. Apply 2-Mean Clustering over pixel set F_{map};
4. $C_1, C_2, nc_1, nc_2, std_1, std_2$=Mean loc., Number of pixels and Standard dev. of Left and Right cluster respectively;
5. d_1, d_2= Euclidean Distance between point C_1 and M_1 and that of between C_2 and M_2 respectively;
6. $d = 0.7 * max(d_1, d_2) + 0.3 * min(d_1, d_2)$;
7. $p_r = \frac{max(nc_1, nc_2)}{min(nc_1, nc_2)}$;[Cluster Point Ratio]
8. $std_r = \frac{max(std_1, std_2)}{min(std_1, std_2)}$;[Cluster Standard Dev. Ratio]
9. $comb_r = 0.8 * p_r + 0.2 * std_r$;
10. $D_{std} = 1 - \frac{d}{\sqrt{std_1^2 + std_2^2}}$;
11. $D_{nc} = 1 - \frac{d}{\sqrt{nc_1^2 + nc_2^2}}$;
12. $S = 0.5 * d + 0.2 * comb_r + 0.15 * D_{std} + 0.15 * D_{nc}$

Back

Algorithm 5.3 $CIOF(knuckle_a, knuckle_b)$

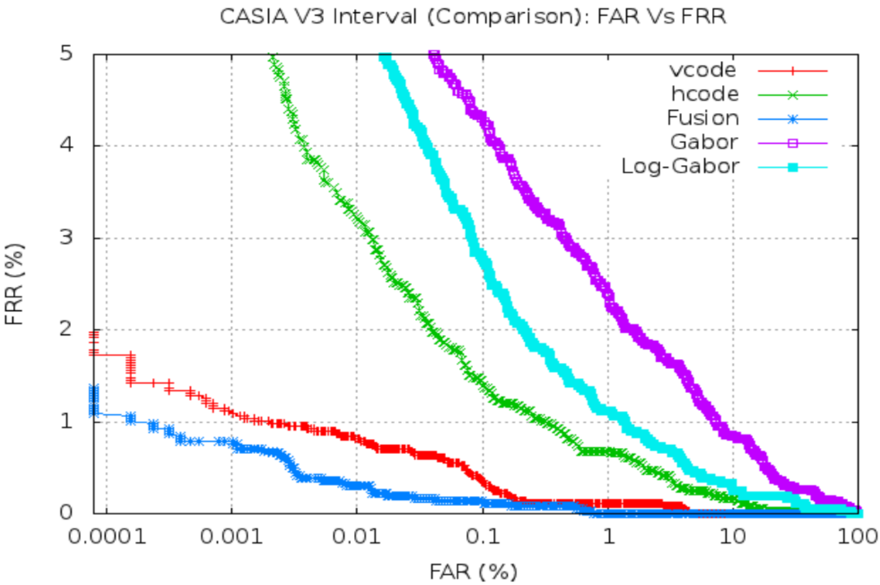
Require:
(a) The *vcode* I_A^v,I_B^v of two knuckleprint images $knuckle_a, knuckle_b$ respectively.
(b) The *hcode* I_A^h,I_B^h of two knuckleprint images $knuckle_a, knuckle_b$ respectively.
(c) N_a^v, N_b^v,N_a^h and N_b^h are the number of corners in I_A^v, I_B^v, I_A^h and I_B^h respectively.

Ensure: Return $CIOF(knuckle_a, knuckle_b)$.

1: Track all the corners of *vcode* I_A^v in *vcode* I_B^v and that of *hcode* I_A^h in *hcode* I_B^h.
2: Obtain the set of corners successfully tracked in *vcode* tracking (*i.e.* stc_{AB}^v) and *hcode* tracking (*i.e.* stc_{AB}^h) that have their tracked position within T_d, their local patch dissimilarity under T_e and also the patch-wise correlation is at-least equal to T_{cb}.
3: Similarly compute successfully tracked corners of *vcode* I_B^v in *vcode* I_A^v (*i.e.* stc_{BA}^v) as well as *hcode* I_B^h in *hcode* I_A^h (*i.e.* stc_{BA}^h).
4: Quantize optical flow direction for each successfully tracked corners into eight directions (*i.e.* at an interval of $\frac{\pi}{8}$) and obtain 4 histograms H_{AB}^v, H_{AB}^h, H_{BA}^v and H_{BA}^h using these four corner sets $stc_{AB}^v, stc_{AB}^h, stc_{BA}^v$ and stc_{BA}^h respectively.
5: For each histogram, out of 8 bins, the bin (*i.e.* direction) which is having the maximum number of corners is considered as the consistent optical flow direction. The maximum value obtained from each histogram is termed as corners having consistent optical flow represented as $cof_{AB}^v, cof_{AB}^h, cof_{BA}^v$ and cof_{BA}^h.
6: $ciof_{AB}^v = 1 - \frac{cof_{AB}^v}{N_a^v}$; [Corners with Inconsis. Optical Flow (*vcode*)]
7: $ciof_{BA}^v = 1 - \frac{cof_{BA}^v}{N_b^v}$; [Corners with Inconsis. Optical Flow (*vcode*)]
8: $ciof_{AB}^h = 1 - \frac{cof_{AB}^h}{N_a^h}$; [Corners with Inconsis. Optical Flow (*hcode*)]
9: $ciof_{BA}^h = 1 - \frac{cof_{BA}^h}{N_b^h}$; [Corners with Inconsis. Optical Flow (*hcode*)]
10: **return** $CIOF(Knuckle_a, Knuckle_b) = \frac{ciof_{AB}^v + ciof_{AB}^h + ciof_{BA}^v + ciof_{BA}^h}{4}$;

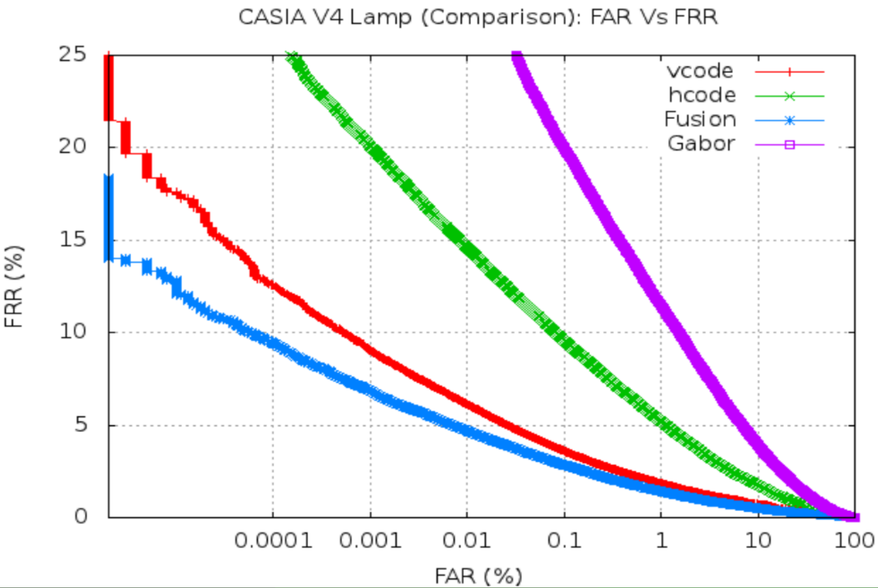
Performance Analysis - Iris(Interval)

Back

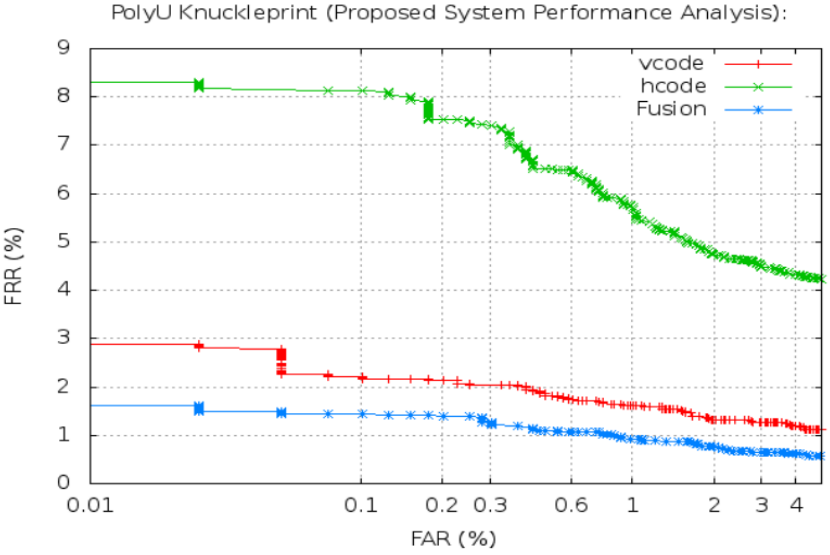


Performance Analysis - Iris(Lamp)

Back

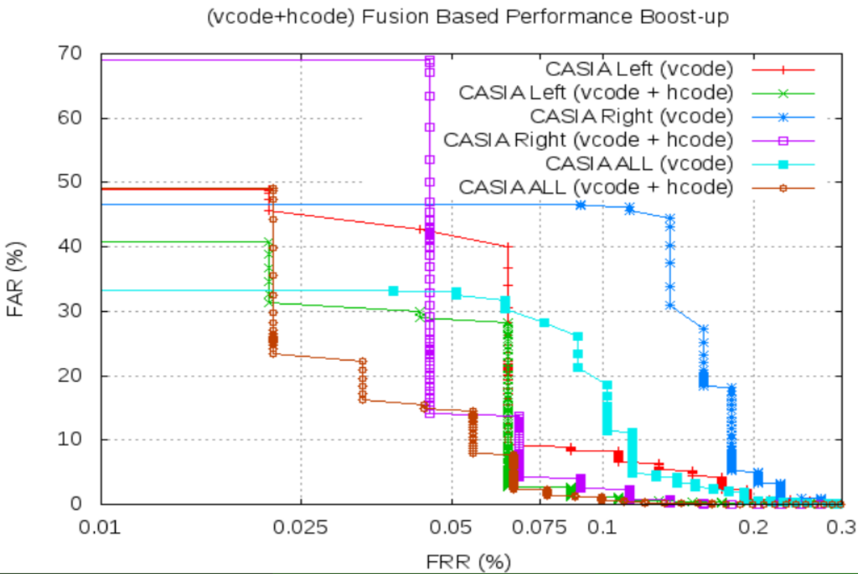


Back


PolyU Knuckleprint (Proposed System Performance Analysis):
vcode
hcode
Fusion
FRR (%)
FAR (%)
0
1
2
3
4
5
6
7
8
9
0.01
0.1
0.2
0.3
0.6
1
2
3
4

Performance Analysis - Palmprint(Casia)

Back



Performance Analysis - Palmprint(PolyU) (I)

Back