Isometric Efficient and Accurate Fourier-SIFT method in Iris Recognition System

Ankush Kumar, and Banshidhar Majhi

Abstract— **Iris is the Optimum Biometric-trait present in Biometrics Security. Our emphasis on this paper is to obtain efficient, fast and robust algorithm set for iris detection. There are number of algorithms proposed for the efficient result but fails due to limitations. We tried in this paper to make, an efficient combination of the best schemes of normalization, corner detection, feature extraction and best matching algorithm available. We have used Modified-Trajkovic Operator (8-neighbours) to detect the corner, in which, the iris image sample is first optimized by sector based normalization into four sectors, this decreases the iris area but the modified 8-neighbours detect the corner accurately. Fourier-SIFT is then used to determine the keypixels with enhanced threshold cutoff and finally Modified-Hausdorff distance (or Gromov-Hausdorff distance) determines the matching algorithm and measures the distance between keypixels of enrolled and scanned iris during matching. The limitations in the above algorithms are rectified in this paper. This process is rigorously checked on CASIA-V3 and MMU iris images.**

Index Terms— **Biometrics, Image edge detection, Image Processing, Iris recognition**

I. Introduction

WHEN talking about Definitude Biometric Security ,Iris comes at the *leading* position. With all leading perfection and appropriateness we are unable to achieve *100% results*, because of lack of optimized set of algorithms involve in the recognition process and the errors present at different steps of the methods used. The efficient methods are available, but not for all sets of iris images and in all environment, like *cooperative* and *non-cooperative* [5]. In this paper we have tried to achieve the maximum accuracy and fastest approach for both cooperative and non-cooperative dataset of iris images. *Trajkovic Operator* [2] is used to find the corner of any image pattern. But it has certain limitations in it:

A. *Anisotropic Response of Operator* .And

B. *False Corners along Diagonal Lines*.

We have corrected these limitations by applying *Diagonal line in pixels* and *SIFT* [1], [6], [13], [14], [16]. These techniques increase the corner detection technique and exact corners and edges are detected. *Scale Invariant Feature Transform* (SIFT) is used to make our approach *invariant* towards *rotation*, *scaling* [7] and *occlusion* [13] by eyelids. We have not only used the SIFT descriptor to detect keypixels, but also used *Fourier-SIFT* based keypixels as a feature keypoints, by applying Fourier transform on each keypixels obtained after SIFT.

SIFT extract features using gradient information only which may not be suitable for iris image. This motivates us to use F-SIFT instead of SIFT and SURF, because of the combination of both Fourier and SIFT together. Sift also shows some false matching if the texture and eyelash ate present as shown in fig. 1a.

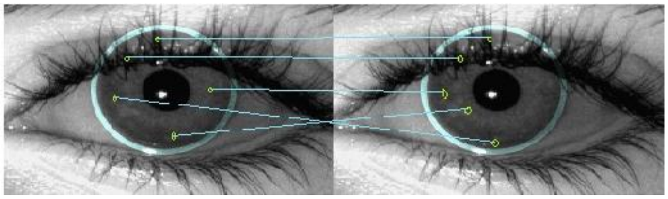

Fig.1a. False Matching in SIFT for non-normalized image.

The first line shows the region of eyelash which cannot be consider as the keypixel for matching. This problem can be solved by good matching algorithm like *Gromov-Hausdorff distance* [9], [10], [17], [18], [19]. It only cares about the interpoint distance between the whole keypixel and isometric EH [1] takes care of rest. Both this method is applied in Modified *Gromov-Hausdorff distance*.

An endeavor has been made in this paper to correct the accuracy rate and the efficiency of iris recognition for real-time applications. The result shows a drastic change with the datasets we have used in this experiment.

II. Modified-Teajkovic Operator (8-Neighbour)

A. Drawback in previous Operators

Teajkovic 4-Neighbours [2], [3], [15], [16], Operator was extremely sensitive to noise and we cannot introduce noise to find the edges and corners. Secondly it is very sensitive to position and rotation invariant situations (*Anisotropic Response*) [23]. *Teajkovic 8-Neighbours* Operator also contains the same problem, since the iris corners are not diagonal as shown in fig 2a. We can easily see the meeting point of L1 and M1 is the locus on the edge of the iris

Ankush Kumar is M Tech Scholar in the Department of Computer Science & Engineering, at National Institute of Technology, Rourkela, Odisha, 769008, INDIA (phone: +917735585378; e-mail: ankuraja@live.com).

Banshidhar Majhi, Senior, is with National Institute of Technology, Rourkela, Odisha 769008 INDIA. He is now with the Department of Computer Science & Engineering, (e-mail: bmajhi@nitrkl.ac.in).

respectively.

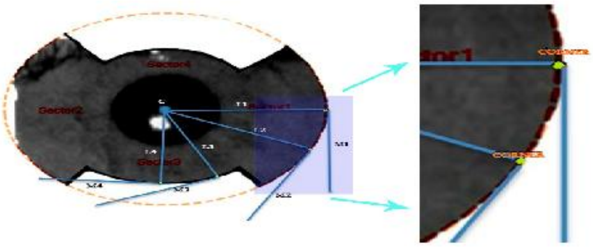

Fig.2a.The Edge locus in modified Teajkovic 8-Neighbours

B. *Teajkovic 8-neighbour Operator*

All 8 corners are considered as the *illumination* [15], [16], points for calculating the corner in the image. The Sliding 8 pixel window determines the edge in the provided image. We took *gray* version of the image as input, with the size of the pixel (i.e. can be calculated from the image size) and the two threshold values 2_1 and 2_2. The algorithm is shown below with different pixel values:

Algorithm 1: Teajkovic 8-neighbour

```
Input: Gray image, Threshold $2_1$ and $2_2$, Scale of image. Image Map (M)
Output: Image With each corner properly detected.

For all pixel (x,y), Find Simple Cornerness measure ($C_M$)
        $C_M(x, y) = \min(r_P, r_Q, r_R, r_S)$
            Where
                $r_P = 2(I_C)^2 + (I_P)^2 + (I_{P'})^2 - 2I_C(I_P+I_{P'})$,
                $r_Q = 2(I_C)^2 + (I_Q)^2 + (I_{Q'})^2 - 2I_C(I_Q+I_{Q'})$,
                $r_R = 2(I_C)^2 + (I_R)^2 + (I_{R'})^2 - 2I_C(I_R+I_{R'})$
                $r_S = 2(I_C)^2 + (I_S)^2 + (I_{S'})^2 - 2I_C(I_S+I_{S'})$

        if $C_M(x,y) \geq 2_1$ mark as Starting corner.
            Initialize Cornerness map M with 0(zero).
                end if.

        for all Starting corner pixel(x,y) :
            a). find location (x', y');
            b). find ($C_M$)
            c). if $C_M(x', y') \leq 2_2$, Skip step d) and e). (NOT CORNER)
            d). Calculate interpixel approximation cornerness measure:

$C_{INTERPIXEL}(x, y) = \left\{ \begin{array}{ll} C_M(x, y), & \text{either } Q_i \geq 0 \text{ or } P_i + Q_i \leq 0 \\ \min\left\{ r_i - \frac{Q_i^2}{P_i} \right\}, & i = 1,2,3,4 \text{ that satisfy } Q_i < 0 \text{ and } P_i + Q_i > 0' \end{array} \right\}$
            Where
                $r_1 = r_P$, $r2 = r_Q$, $r3 = r_R$, $r4 = r_S$
                $Q_1 = (I_Q - I_P)(I_P - I_C) + (I_{Q'} - I_{P'})(I_{P'} - I_C)$ same for 2,3 and 4.
                $P_1 = r_Q - r_P - 2Q_1$    similarly for 2, 3, and 4.

            e).if $C_{INTERPIXEL}(x', y') \leq 2_2$ LEAVE IT
                    else
                        Set M (x', y') to $C_{INTERPIXEL}(x', y')$.

    Apply Non-maximum suppression on M (x', y') for local maxima.

    Finally all Non-zero points are Corners.
```

In *Teajkovic Operator (8-neighbour)* the number of false corners is very less in numbers than that of 4-neighbor and the rotation invariant problem is solved by applying SIFT invariant in it. These limitations led to the generation of new and improved approach called *Modified –Teajkovic Operator* [3].

C. *Modified-Teajkovic Operator*

The sliding window, diagonal lines and *rotation invariant* are three problems occurred in 8-neighbour operator, which motivates us to design the new approach called *Modified – Teajkovic Operator* [23]. In sliding window, we took not only the pixel matrix but also the angular component at an angle of 45°. This led to the smoothening of the diagonals. The fig.2b shows the corrected sliding window with the *angular component*.

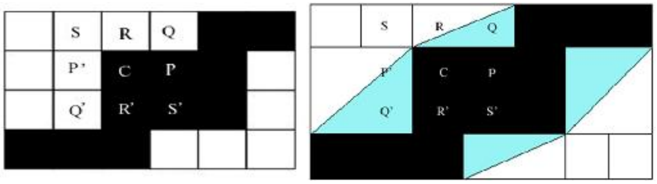

Fig.2b. (left) Normal Sliding Window in Teajkovic 8-Neighbours (Right) corrected Sliding Window in Teajkovic 8-Neighbours

III. Efficient Sector Normalizations

Inner pupil circle is not fixed; it changes as per the *illumination* sources condition. Thus we see the variation of the inner pupil radii, whereas the outer circle remains fixed. *Normalization* means to shell out the minor deformation present in the sectors. After finding the optimal pupil radii and outer edge of the iris from *Modified-Teajkovic operator*, it's time to get rid of the eyelash. The problem of occlusions can be solved by Efficient Sector *Normalization* [13], [1], [7], [8], in better way. The following Steps remove the eyelash present in the iris:

A) Iris Localization: The Center of pupil can be determined by *localization of iris* and *pupil centers* at (x_0, y_0). With reference to the center we can divide whole iris image into *four* sectors. Then the removal of unwanted noise i.e. eyelids. The coordinates of the pupil and iris are given by:

$$(x_p(\theta), y_p(\theta)) = x_0 + r_p \cos(\theta), y_0 + r_p \sin(\theta).$$
$$(x_i(\theta), y_i(\theta)) = x_0 + r_i \cos(\theta), y_0 + r_i \sin(\theta) \qquad \textbf{(1)}$$

Where (x_p, y_p) are the points lying on the pupil and (x_i, y_i) are the points lying on iris outer boundary, with center (x_0, y_0).

B*) Sector Division:* Once the proper iris is obtained, it is now time for removing eyelids by *sector division*. Whole iris is divided into *four* sectors i.e. *Sector1*, *sector2*, *sector3* and *sector4*. *Sector1* and *Sector2* are nearly same in dimension with very little *occlusion*, θ range of sector 1 is [-65°, 45°], *Sector2* ranges from [135°,245°], *Sector3* range is from [245°,295°] and finally the θ range of *Sector4* is [45°,135°]. The modification is done in the upper and lower sectors, only for *noise* free result.

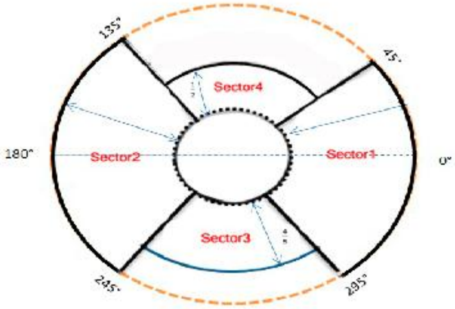

Fig.3a. Representation of all Sectors in redial and Angular form.

The upper Sector (*Sector4*) is chopped in half, basically to remove the eyelids. The lower Sector (*Sector3*) is also clipped in 4/5. Both can be represented as:

$$D_4 = (r_i - r_p)/2$$

$$D_3 = [(r_i - r_p) - 0.2*(r_i - r_p)] \qquad (2)$$

The equation 2 gives the corrected size of the normalized iris strip as shown in the fig 3a. The correct angle can remove most of the *noise* present in the image. It is done to avoid the generation of keypixels in the noise (i.e. *eyelid*) region. It may lead to false matching (since the keypixel in eyelid region is basically from *black eyelash* and other people can also have the *black eyelash* keypixel which lead to the matching error).

. The normalization improves the accuracy about 20% from non-normalized image.

IV. Keypixel Detection in Fourier SIFT

Scale Invariant Feature Transform (SIFT) [1], [6], [7], [13], [14], [16], is well known keypixel descriptor for *image recognition*. Previously SIFT is combined with *Gabor wavelet* [4], [8], for the *feature* extraction and method is tested using *frontal* and *off angle* images of iris. But in that method *Fourier transforms* [14], [6], [16], (*Phase Component*) is used to find the texture present in the image, because of the dependency of *amplitude* on extraneous factors.

In this paper Fourier-SIFT is used to find the keypixels present in the iris image. Fourier-SIFT is the combination of *Fourier Transformation* with *SIFT*. Both can be paired using *Phase-Only Correlation (POC*). This Combination of Fourier and SIFT gives better result than all previously used techniques. This approach is best for both *cooperative* and *non-cooperative* [5] iris images. The process of Fourier-SIFT can be broken into *three* steps:

Step I: Finding Keypixels using SIFT: - SIFT (Scale Invariant Feature Transform) is a local *Feature Extraction Technique* not only for its *scaling* benefits but also for *occlusion* and *illumination conditions*. SIFT is very c*omplex and involved algorithm*, so for simplicity we have divided it into five steps. *First step* is converting our image in required scale (called *Scale space*). The iris image is progressively blurred using *Gaussian blur* [6], [7], [8], [11], and resize to half of the original image size. These blurred images are called the *Octaves*. *"Blurring" is represented as:*

$$L(x,y.\sigma) = G(x,y,\sigma) * I_m(x,y)$$

The *Gaussian* blur can be shown as:

$$G(x, y, \pm) = \frac{1}{2 \circledR \pm^{2}} e^{-(x^2+y^2)/2\pm^2}$$

Where: *$x' = x \cos\theta + y \sin\theta$ and $y' = -x \sin\theta + y \cos\theta$, L is the blurred image, G is Gaussian Blur Operator [8], I_m is an image, (x,y) are the location Coordinates, σ is the scale of blurring (amount of bluring).*

Once *scaling* is done, *Second step* is to find the DoG (Difference of Gaussian) instead of *Laplacian of Gaussian (LoG)*. *Laplacian* is the second order derivative of iris image. The DoG is shown mathematically as:

$$D(x, y.\sigma) = L(x, y.\kappa\sigma) - L(x, y.\sigma)$$

LoG will locate the fine *corner* and *edge* present in image, which is good for finding the keypixels. But due to the *extremely sensitive characteristics* of the second order derivative towards *noise*, we can't choose LoG. The solution is DoG which gives same result as that of LoG but without noise, which provide *accuracy* to the method. During this process we simply subtract the *preceding Gaussian blurred* image to the next image. *Third step* is to determine keypixels. Finding keypixels consist of *locating the maxima/ minima* in DoG image by selecting the desired pixel P_0 and *comparing it with its nearest 26 pixels*. We compare the *illumination* conditions and the maximum and minimum pixels are calculated. The points observed in this process are the *approximated* maximum and minimum. As the maxima or minima never lies exactly on the pixel. It lies somewhere between the pixels, so we must locate its position mathematically. Then sub-pixel maxima or minima are calculated by t*aylor expression* [1], [6], [7], of iris image around P_0.*Taylor expression* is given as:

$$M(P) = M + \frac{\partial M^T}{\partial x} + \frac{1}{2} p^T \frac{\partial^2 M}{\partial x^2} p$$

At this point we have a lot of keypixels as an output. *Fourth step* is to remove the *flat* region. It will be then tested on *harris-corner detector* [15], [16], [23], which will led to two outputs (i.e. gradients). Then based on these values *edge*, *corner* and *flat* region can be determined. An *edge* consist of *huge* gradient (only perpendicular to edge) other will be small, a *corner* consist of both gradient as big gradients. The *flat* region consists of both small gradients. Since corner and edge are invariant points we will consider only those points and filter the flat points. Then we assign the *orientations* around these keypixels. We have to collect g*radient direction* and g*radient magnitude* around these keypixels. It can be calculated using the formula:

$$m(x,y) = \sqrt{(L(x+1,y) - Lx-1,y)^2 + (L(x,y+1) - L(x,y-1))^2}$$

$$^{-}(x,y) = \tan^{-1}\left(\frac{(L(x,y+1) - L(x,y-1))}{(L(x+1,y) - Lx-1,y))}\right)$$

Then a histogram is created by dividing 360° in 36 bars each of 10° as shown in the figure 4a.

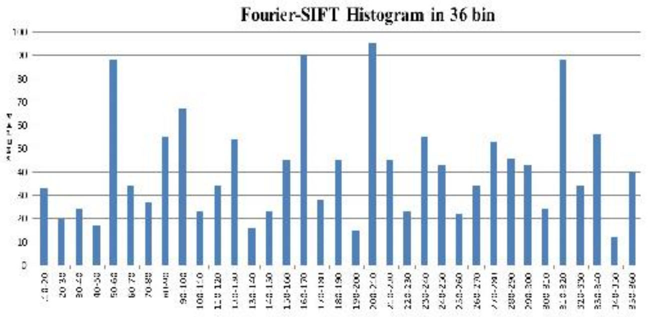

Fig.4a. Graph showing Histogram of the Fourier SIFT.

Fifth step is to create SIFT feature with the generated keypixels from last step. A *16x16* window is taken around the keypixel and it is further divided into *4 x 4* windows. Within all 4 x 4 window *gradient magnitude* and *gradient orientation* are calculated. These are further kept in 8 bins histogram [0°-45°], [45°-90°], [90°-135°], [135°-180°], [180°-225°], [225°-270°], [270°-315°] and [315°-0°]. The amount added to the bin equals to *magnitude of gradient*. The value in the bin can be kept according to their *original magnitude* values. Then a *histogram* is generated. The bar whose value is greater than 85% is taken as the keypixels. In the above diagram only four points are taken, since the amount also depends upon *distance from the keypixel,* thus the gradient far away from keypixel will also be neglected. It can be better explained by Gaussian Weighting Function.

$$G_{WEIGHT} = (m(x,y) \times ^{-}(x,y)) * H_{MEAM}$$

Where G_{WEIGHT} is Gaussian Weighting Function and H_{MEAN} is the 8-Bin Histogram.

Step II: Fourier Transform of the Keypixels: - We have now calculated the location of keypixels .The window size taken in *step 1* is 16 at every keypixel whose center is at *(x, y)* with its orientation (θ) and magnitude *m*. The *global frequency* data can be calculated by Fourier transform on the calculated keypixels. Each descriptor around keypixel is calculated by the complex equation as:

$$F_{Key\ -descriptor} = \frac{1}{256} \sum_{n_1=(x-8)}^{x+8} \sum_{n_2=(y-8)}^{y+8} I(n_1, n_2) e^{-i2® \left(\frac{n_1 u}{16} + \frac{n_2 v}{16} \right)}$$

Where

u>0 and $(x-8) \le u \le (x+8)$,

v=0 and $(y-8) \le v \le (y+8)$

(u, v) are the phase component and other are the amplitude component. If we take the window size as W then we can also get the generalized equation by simply changing the numerals values.

This will give the output as in form of *amplitude* component and *phase* component.

Step III: Pairing Using Phase-Only Correlation (POC) [22]*: -* From previous step we get two output A_i *(u, v)* and θ_i *(u, v)* as amplitude and Phase Component of the Fourier Transform of ith keypixel. The Phase-Only Correlation (POC) [22] function determine the similarity between two keypixels by relating the ith coordinate with jth coordinate in complex conjugate. It can be calculated by two methods, *first*, by pairing the phase information of ith keypixel with jth keypixel in the probe image using the POC function ($P_{(ij)}$). POC function ($P_{(ij)}$) is completely inverse of the *Fourier* Keypixel Descriptor.

$$P_{(ij)} = F_{keydescriptor}$$

POC function gives a sharp spike at the similar texture and similar keypixel region which is used to correct the error present in SIFT. However, the spike height deceases for the two dissimilar regions.

V. Matching Algorithms

We have extracted the final keypixel from the iris specimen and saved it in the database. During *authentication* we match the pattern observed from the person with the existing pattern stored during *enrollment*. The matching should be accurate and effective with more FRR (*False Rejection Ratio*). We choose *Gromov-Hausdorff distance* for our matching process. Taking *Euclidean isometrics Space* in consideration *Gromov-Hausdorff* measures *the longest distance of a set to the shortest point in the other set* (Felix Hausdoroff**,** (1868-1942)).

$$¢_{H}(X,Y)\{Hausdoroff\} = \underset{dis(x)\in A}{Max}\{\underset{dis(y)\in B}{Min}\{Dis(x,y)\}\}$$

Where x and y are any point of cluster point set A and B, and (x, y) is any metric between these points. Now in the *Gromov-Hausdorff* distance we introduce the complex matric space P and Q.

$$d_{GH}(P,Q)\{Gromov - Hausdoroff\} = \underset{Z,f,g}{Dis}\{¢_{H}(f(P), g(Q))\}$$

Where f: P → Z and g: Q → Z are *isometric* distance into metric space (Z, d).

This measures the *minimum* distance and match the exact keypixel which is calculated after Fourier-SIFT.

VI. Experimental Results

A. Databases and iris images

We have taken the publicly available MMU [21] iris, the *non-cooperative* iris image like CASIA-irisV3 [20] (all) images. MMU [21] consist of a two group MMU1 and MMU2, MMU1 contains a total of 450 iris images of 90 different classes, the size of the image is 320*240 in BMP format.CASIA-irisV3 consist of different database section and captured in the indoor environment. We have performed our test on *Lamp*, *interval* and *twins* databases only. Most of the images were taken in two sessions within an interval of one month. It contains a total of 2655 images of 249 subjects of both left and right eye. Each section contains different images and different classes. The images of CASIA-irisV3 and MMUI are shown in fig. 6a.

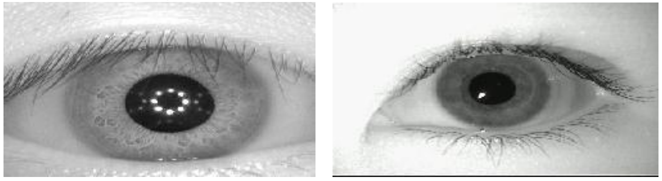

Fig. 6a.CASIA and MMU iris image database samples.

B. *Results*

The results are obtained on the basis of comparison of the results with the traditional SIFT and our new approach. The *false matching* in SIFT was nearly 22-100 out of million keypixels and the *Fourier-SIFT* is having only 4 out of million, that also depend upon bad *illumination* condition and bad quality (in 4th or 5th octaves). FRR of both the iris database is nearly 0.00 %. The accuracy of Fourier-SIFT is more than 99.0 % (This shows the accuracy and efficiency of the algorithm set).

TABLE I
ACCURACY, FRR, AND FAR (IN %) FOR CASIA-V3 AND MMU

	CASIA-V3			MMU		
	FRR	FAR	ACC	FRR	FAR	ACC
SIFT	9.43	20.18	85.1	10.2	13.11	90.2
Fourier-SIFT	0.00	2.01	98.99	0.3	3.22	99.10

Table 1 shows the accuracy, FRR and FAR of two iris database. The accuracy associated with **CASIA v3** [20] was 85.1 % for SIFT and 98.99 % in Fourier-SIFT. Similarly in MMU [21] also accuracy observed is more than 99% (i.e. 99.10%). *Phase Only Correlation (POC)* [22] distinguishes between the same texture of the iris image and thus increasing the efficiency. Accuracy is measured by taking the average of FAR and FRR from 100.

The *ROC (Receiver Operating Characteristics)* [1], [11], 12] curve of *Fourier-SIFT* is very less and hence more accurate.

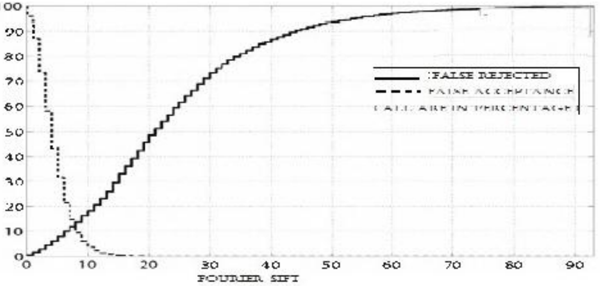

Fig.6b. Accuracy Graph Showing FA and FR for MMU iris database

The graph in fig. 6b shows the total acceptance of imposter and genuine iris images and generated in matlab. The entire database is rounded into 100 %. These shows only 40 false are rejected out of 100% and less than 9 are wrongly accepted. The matching algorithm is completely accurate and time efficient. There was only 0.03% wrong match during identification. These percentages consist of hardware (device) error also.

VII. Conclusion

We finally conclude that SIFT alone is not enough for feature detection process because it is not precise in *illumination change*. Our presented algorithm set performs with accuracy more than 99%. The *Fourier transformation* of SIFT increases the Phase-only correlation (POC) [22], which helps in texture distinction. We have reform all limitations present in the applied algorithm, which helps in the improvement of iris recognition with great accuracy of 99.9. The similar approach was also applied in IIT Delhi database and produces 100% result.

The further improvement of the feature can be achieved by applying Time Frequency Representation (TFR) in PCO. The POC can be taken in 2-D and 3-D for better results. These changes will only improve a little but only a little is required for cent percent perfection.

References

[1] Hunny Mehrotra, Badrinath G.S., Banshidhar Majhi, "*An Efficient Iris Recognition using local feature descriptor*," ICIP 2009, pp. 1957–1960.

[2] M. Trajkovic and M. Hedley. *"Fast Corner Detection. Image and Vision Computing"*, Vol. 16(2), pp. 75-87, 1998.

[3] C. Harris and M. Stephens. "*A Combined Corner and Edge Detector"*. Proc. Alvey Vision Conf., Univ. Manchester, pp. 147-151, 1988.

[4] J. Daugman. "*The importance of being random: statistical principles of iris recognition"*. Pattern Recognition, 36(2):pp. 279 – 291, 2003.

[5] H. Proenca and L.A. Alexandre. "*A classification approach using multiple signatures"*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 29(4):pp. 607 –612, 2007.

[6] Lowe, D.G." *Distinctive image features from scale-invariant keypoints"*. IJCV 60, pp. 91–110, 2004

[7] Ke, Y., Sukthankar, R. "*PCA-SIFT: A more distinctive representation for local image descriptors*". CVPR. Volume 2. pp. 506–513, 2004

[8] J. Daugman. "*New methods in iris recognition*". IEEE Transactions on Systems, Man, and Cybernetics, Part B, 37(5):pp. 1167–1175, 2007.

[9] Marie-Pierre Dubuisson and Anil K. Jain, "*A Modifed Hausdor Distance for Object Matching*," Proc. International Conference on Pattern Recognition, Jerusalem, Israel, pp 566–568, 1994.

[10] Normand Grégoire and Mikael Bouillot, "*Hausdorff distance between olygon"*. *[*Online*]*http://cgm.cs.mcgill.ca/~godfried/teaching/cg-projects/98/normand/main.html.

[11] H. Mehrotra, B. Majhi, and P. Gupta. "*Annular iris recognition using SURF*". Pattern Recognition and Machine Intelligence, volume 5909 of Lecture Notes in Computer Science, Springer, pp. 464–469, 2009.

[12] S. Bakshi, H. Mehrotra, and B. Majhi. "*Real-time iris segmentation based on image morphology*".International Conference on Communication, Computing & Security, ACM, pp. 335–338. 2011.

[13] H. Mehrotra, G.S. Badrinath, B. Majhi, and P. Gupta. "*An efficient iris recognition using local feature descriptor*". 16th IEEE International Conference on Image Processing (ICIP), pp. 1957 –1960, 2009.

[14] C. Belcher and Y. Du. "*Region-based SIFT approach to iris recognition*". Optics and Lasers in Engg. 47(1):pp. 139 – 147, 2009.

[15] Z.Zheng, H.Wang and EKTeoh. "*Analysis of Gray Level Corner Detection*". Pattern Recognition Letters, Vol. 20, pp. 149-162, 1999.

[16] H. P. Moravec. "*Towards Automatic Visual Obstacle Avoidance*". Proc. 5th International Joint Conf. on A. I. , pp. 584-590, 1977.

[17] Facundo M´emoli. "*Gromov-Hausdorff distances in Euclidean spaces"*,[unpublished].

[18] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge. "*Comparing images using the hausdorff distance*". IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 15, 1993.

[19] N. J. Kalton and M. I. Ostrovskii. "*Distances between branch spaces*". Forum Math., pp. 17–48, 1999.

[20] CASIA Database. *http://www.cbsr.ia.ac.cn/english/IrisDatabase.asp*.

[21] MMU1 &MMU2 iris database, *http://pesona.mmu.edu.my/~ccteo/*.

[22] K. Ito, H. Nakajima, K. Kobayashi, T. Aoki, and T. Higuchi.*"A fingerprint matching algorithm using phase-only correlation". IEICE Transactions Fundamentals, 87-A (3):pp. 682–691, 2004.*

[23] Donovan Parks and Jean-Philippe Gravel, [Online] *http://kiwi.cs.dal.ca/~dparks/CornerDetection/trajkovic.htm*.

[24] Zhang S., Tian, Q., Lu K., Huang Q. "*Edge-SIFT: Discriminative Binary Descriptor for Scalable Partial-Duplicate Mobile Search*," Image Processing, IEEE Transactions on , pp. 99, 2013

[25] Bakshi S., Mehrotra H., Sa P.K., "*Score level fusion of SIFT and SURF for iris*," (ICDCS), 2012 International Conference , pp. 527-531, 2012