# Identification of a Person by Palm Geometry Using Invariant Features

**Himanshu Maurya, Shikha Maurya, B. C. Sahana**

***Abstract*— Feature extraction is one of the main topics in Computer Vision. This paper presents the extraction of features of interest from two or more images of the same and different objects and the matching of these features in adjacent images. Each of these feature vectors is supposed to be distinctive and invariant to any scaling, rotation or translation of the image. It uses the SIFT (Scale Invariant Feature Transform) technique for feature extraction from the image. The paper describes our own implementation of the SIFT algorithm and highlights potential direction for future research.***

***Index Terms*—Biometric, Features, Matching, SIFT.**

## I. INTRODUCTION

Biometrics make use of the physiological or behavioral characteristics of people such as fingerprint, iris, face, palmprint, gait, and voice, for personal identification, which provides advantages over non biometric methods such as password, PIN, and ID cards. The most widely used biometric feature is the fingerprint and the most reliable feature is the iris. However, it is very difficult to extract small unique features (known as minutiae) from unclear fingerprints and the iris input devices are very expensive. Other biometric features, such as the face and voice, are less accurate and they can be mimicked easily [1].

Recently, a novel hand-based biometric feature, palmprint, has attracted an increasing amount of attention. Palmprints are believed to have the critical properties of universality, uniqueness, and collectability for personal authentication [2]. What's more, palmprints have some advantages over other hand-based biometric technologies, such as fingerprints and hand geometry. Palms are large in size and contain abundant features of different levels, such as creases, palm lines, texture, ridges, delta points and minutiae. Faking a palmprint is more difficult then faking a fingerprint because the palmprint texture is more complicated, and one seldom leaves his/her complete palmprint somewhere unintentionally. Also, compared to fingertips, palms are more robust to damage and dirt. What is more, low-resolution imaging can be employed in the palmprint recognition based on creases and palm lines, making it possible to perform real time image preprocessing and feature extraction. Among the four common hand-based biometric features: fingerprint [3], hand geometry [4], palm vein and palmprint [2]. The palmprint is believed to be able to achieve the authentication accuracy comparable to that of the fingerprint, and is higher than the accuracies of the hand geometry recognition.

**Manuscript received May, 2013.**

**Himanshu Maurya**, Electronics & Communication Engineering, National Institute of Technology, Patna, India.

**Shikha Maurya**, Computer Science Engineering, ABVIIITM, Gwalior, India.

**Prof. Bikas Chandra Sahana**, Electonics & Communication Engineering, National Institute of Technology, Patna, India.

Texture and palm lines are the most clearly observable palmprint features in low resolution images, and thus have attracted most research efforts. In the palm line based methods, the slope, intercept, inclination or orientation of palm lines are used as features. In texture based approaches, texture features are extracted by filtering palmprint images using filters such as the Gabor filter, the ordinal filter, or the wavelet. For the purpose of extending the palmprint authentication technology, we propose a novel representation and matching method for palmprint incorporating SIFT (Scale Invariant Feature Transformation) [5]. SIFT has already been widely used in generic object detection problems. Recently, its application in face authentication [6] and fingerprint verification [7] has also been studied.

## II. PREVIOUS WORK

David G. Lowe [5] used the SIFT features for object recognition. According to their work, the invariant features extracted from images can be used to perform reliable matching between different views of an object or scene. The features have been shown to be invariant to image rotation and scale and robust across a substantial range of affine distortion, addition of noise, and change in illumination. The approach is efficient on feature extraction and has the ability to identify large numbers of features. Our implementation focuses on deriving SIFT features from an ear image and trying using these features to perform human identification.

## III. SIFT (SCALE INVARIANT FEATURE TRANSFORM)

This is the method which we are using to find out distinctive and invariant features. The algorithm has four stages.

### *A. Scale Space Extrema Detection*

This is the stage where the interest points, which are called keypoints in the SIFT framework, are detected. For this, the image is convolved with Gaussian filters at different scales, and then the difference of successive Gaussian-blurred images is taken. Key points are then taken as maxima/minima of the Difference of Gaussians (DoG) that occur at multiple scales. Specifically, a DoG image $D(x, y, \sigma)$ is given by,

$$D(x, y, \sigma) = L(x, y, k_i\sigma) - L(x, y, k_j\sigma) \quad (1)$$

Where $L(x, y, k\sigma)$ is the original image $I(x, y)$ convolved with the Gaussian blur $G(x, y, k\sigma)$ at scale kσ, i.e.

$$L(x, y, k\sigma) = G(x, y, k\sigma) * I(x, y) \quad (2)$$

Where,

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{\frac{-(x^2+y^2)}{2\sigma^2}} \quad (3)$$

Hence a DoG image between scales $k_i\sigma$ and $k_j\sigma$ is just the difference of the Gaussian-blurred images at scales $k_i\sigma$ and $k_j\sigma$. For scale-space extrema detection in the SIFT algorithm, the image is first convolved with Gaussian-blurs at different scales. The convolved images are grouped by octave, and the value of $k_i$ is selected so that we obtain a fixed number of convolved images per octave. Then the Difference-of-Gaussian images are taken from adjacent Gaussian-blurred images per octave.

The approach we use in our implementation is first, the initial image, I, is convolved with a Gaussian function, $G_0$ , of width $\sigma_0$ . Then we use this blurred image, $L_0$ , as the first image in the Gaussian pyramid and incrementally convolve it with a Gaussian, $G_i$ , of width $\sigma_i$ to create the $i_{th}$ image in the image pyramid, which is equivalent to the original image filtered with a Gaussian, $G_k$ , of width $k\sigma_0$ . The effect of convolving with two Gaussian functions of different widths is most easily found by converting to the Fourier domain, in which convolution becomes multiplication i.e.

$$G_{\sigma_i} * G_{\sigma_0} * f(x) = \hat{\mathrm{G}}_{\sigma_i}\hat{\mathrm{G}}_{\sigma_0}\mathrm{f} \quad (4)$$

The Fourier transform of a Gaussian function, $e^{ax^2}$ is given by,

$$F_x\left[e^{ax^2}\right](t) = \sqrt{\frac{\pi}{a}}\, e^{\frac{-\pi^2(t^2)}{a}} \quad (5)$$

Substituting this into equation (4) and equating it to convolution with a single Gaussian of width $k\sigma_0$we get,

$$e^{-t^2{\sigma_i}^2} e^{-t^2{\sigma_0}^2} = e^{-t^2k^2{\sigma_0}^2} \quad (6)$$

Performing the multiplication of the two exponentials on the left of this equation and comparing the coefficients of $-t^2$ gives,

$${\sigma_i}^2 + {\sigma_0}^2 = k^2{\sigma_0}^2 \quad (7)$$

Therefore,

$$\sigma_i = \sigma_0\sqrt{k^2 - 1} \quad (8)$$

This subtle point is not made clear by David G. Lowe [5], and it is important because after subsampling of the low-passed filtered images to form the lower levels of the pyramid we no longer have access to the original image at the appropriate resolution, and so we cannot filter with $G_k$ directly. Each octave of scale space in divided into an integer number, $s$, of intervals and we let $k = 2^{1/s}$. We produce $s+3$ images for each octave in order to form $s+2$ difference of Gaussian (DoG) images and have plus and minus one scale interval for each DoG for the extrema detection step. In this experiment, we set $s$ to 3, following on from experimental results in [1] that suggests that this number produces the most stable key-points.

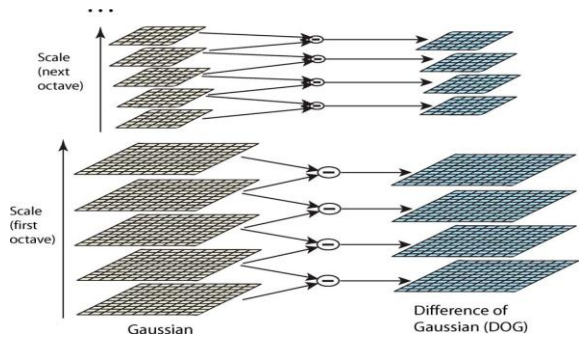


Figure 1. Creation of scale-space and difference of Gaussian function

Once DoG images have been obtained, key points are identified as local minima/maxima of the DoG images across scales. This is done by comparing each pixel in the DoG images to its eight neighbors at the same scale and nine corresponding neighboring pixels in each of the neighboring scales. If the pixel value is the maximum or minimum among all compared pixels, it is selected as the candidate keypoint.

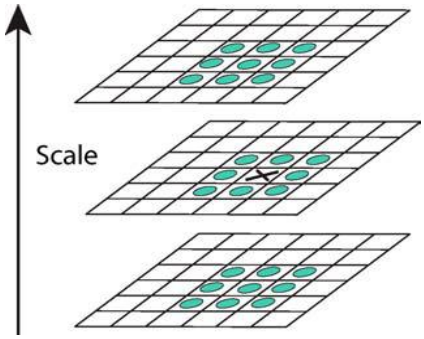


Figure 2. The pixel marked with 'X' is compared with its 26 neighbors in 3 × 3 regions at the current and adjacent scales

### *B. Keypoint Localization*

After scale space extrema are detected the SIFT algorithm discards low contrast key points and then filters out those located on edges. Scale space extrema detection produces too many keypoint candidates, some of which are unstable. The next step in the algorithm is to perform a detailed fit to the nearby data for accurate location, scale, and ratio of principal curvatures. This information allows points to be rejected that have low contrast (and are therefore sensitive to noise) or are poorly localized along an edge.

#### *1) Interpolation of nearby data for accurate position*

First, for each candidate keypoint, interpolation of nearby data is used to accurately determine its position. The approach calculates the interpolated location of the maximum, which substantially improves matching and stability. The interpolation is done using the quadratic Taylor expansion of the Difference-of-Gaussian scale-space function $D(x, y, \sigma)$ with the candidate keypoint as the origin. This Taylor expansion is given by,

$$D(X) = D\frac{\partial D^T}{\partial X}X + \frac{1}{2}X^T\frac{\partial^2 D}{\partial X^2}X \quad (9)$$

Where D and its derivatives are evaluated at the candidate keypoint and $X = (x, y, \sigma)$ is the offset from this point. The location of the extremum, is determined by taking the derivative of this function with respect to x and setting it to zero. If the offset is larger than 0.5 in any dimension, then that is an indication that the extremum lies closer to another candidate keypoint. In this case, the candidate keypoint is changed and the interpolation performed instead about that point. Otherwise the offset is added to its candidate keypoint to get the interpolated estimate for the location of the extremum.

#### *2) Discarding low contrast keypoints*

To discard the keypoints with low contrast, the value of the second-order Taylor expansion $D(X)$ is computed at the offset. If this value is less than 0.03, the candidate keypoint is discarded. Otherwise it is kept, with final location and scale σ.

### *3) Eliminating edge responses*

The DoG function will have strong responses along edges, even if the candidate keypoint is unstable to small amounts of noise. Therefore, in order to increase stability, we need to eliminate the keypoints that have poorly determined locations but have high edge responses.

For poorly defined peaks in the DoG function, the principal curvature across the edge would be much larger than the principal curvature along it. Finding these principal curvatures amounts to solving for the eigen values of the second-order Hessian matrix, H,

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \tag{10}$$

The eigen values of H are proportional to the principal curvatures of D. It turns out that the ratio of the two eigen values, say α is the larger one, and β the smaller one, with ratio $r = \alpha/\beta$, is sufficient for SIFT's purposes. The trace and determinant of H are given by,

$$T_r(H) = D_{xx} + D_{yy} = \alpha + \beta \tag{11}$$

$$Det(H) = D_{xx} D_{yy} - \left(D_{xy}\right)^2 \tag{12}$$

Then

$$\frac{T_r(H)^2}{Det(H)} = \frac{(\alpha+\beta)^2}{\alpha\beta} \tag{13}$$

Since, $\alpha = r\beta$, therefore,

$$\frac{T_r(H)^2}{Det(H)} = \frac{(r\beta+\beta)^2}{r\beta^2} = \frac{(r+1)^2}{r} \tag{14}$$

which depends only on the ratio of the eigen values rather than their individual values. Ratio is minimum when the eigen values are equal to each other. Therefore the higher the absolute difference between the two eigen values, which is equivalent to a higher absolute difference between the two principal curvatures of D. We use $r = 10$, which eliminates keypoints that have a ratio between the principal curvatures greater than 10.

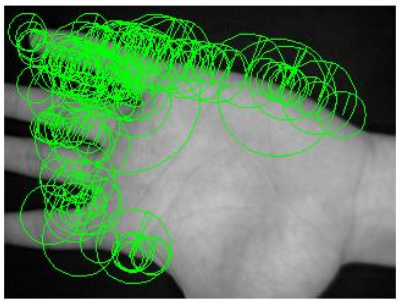

Figure 3. (a) The initial 137 keypoints location at maxima and minima of the difference of Gaussian function

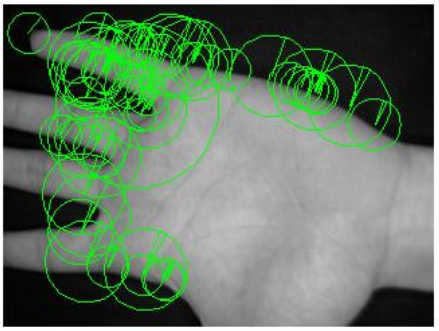

Figure 3. (b) After applying a threshold on low contrast 87 keypoints remain

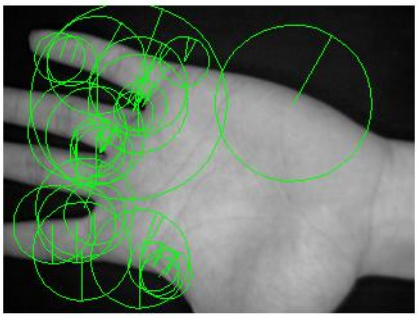

Figure 3. (c) Final 37 keypoints remain by eliminating edge responses

### *C. Orientation assignment*

In this step, each keypoint is assigned one or more orientations based on local image gradient directions. This is the key step in achieving invariance to rotation as the keypoint descriptor can be represented relative to this orientation and therefore achieves invariance to image rotation. First, the Gaussian-smoothed image $L(x, y, \sigma)$ at the keypoint's scale σ is taken so that all computations are performed in a scale-invariant manner. For an image sample, $L(x, y)$ at this scale, the gradient magnitude $m(x, y)$, and orientation $\theta(x, y)$, are pre-computed using pixel differences.

$$m(x,y) = \sqrt{\left(L(x+1,y) - L(x-1,y)\right)^2 + \left(L(x,y+1) - L(x,y-1)\right)^2} \tag{15}$$

and,

$$\theta(x,y) = \tan^{-1}\frac{\left(L(x,y+1) - L(x,y-1)\right)}{\left(L(x+1,y) - L(x-1,y)\right)} \tag{16}$$

The magnitude and direction calculations for the gradient are done for every pixel in a neighboring region around the keypoint in the Gaussian-blurred image L. An orientation histogram with 36 bins is formed, with each bin covering 10 degrees. Each sample in the neighboring window added to a histogram bin is weighted by its gradient magnitude and by a Gaussian-weighted circular window with a σ that is 1.5 times that of the scale of the keypoint. The peaks in this histogram correspond to dominant orientations. Once the histogram is

filled, the orientations corresponding to the highest peak and local peaks that are within 80% of the highest peaks are assigned to the keypoint. In the case of multiple orientations being assigned, an additional keypoint is created having the same location and scale as the original keypoint for each additional orientation.

### *D. Keypoint Descriptor*

Previous steps found keypoint locations at particular scales and assigned orientations to them and ensured invariance to image location, scale and rotation. Now we compute descriptor vectors for these keypoints such that the descriptors are highly distinctive and partially invariant to the remaining variations, like illumination, 3D viewpoint, etc. The feature descriptor is computed as a set of orientation histograms on (4 x 4) pixel neighborhoods. The orientation histograms are relative to the keypoint orientation and the orientation data comes from the Gaussian image closest in scale to the keypoint's scale, the contribution of each pixel is weighted by the gradient magnitude, and by a Gaussian with σ equal to one half the width of the descriptor window. Histograms contain 8 bins each, and each descriptor contains a 4x4 array of 16 histograms around the keypoint. This leads to a SIFT feature vector with (4 x 4 x 8 = 128 elements). This vector is normalized to enhance invariance to changes in illumination.

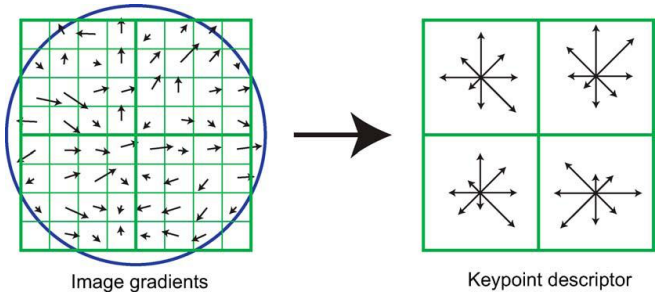


Figure 4. A 2 × 2 descriptor

## IV. RESULTS

Results are obtained by applying the algorithm on the CASIA-palmprintV1.

### *A. Results showing SIFT keypoints on palm images of two different subjects*

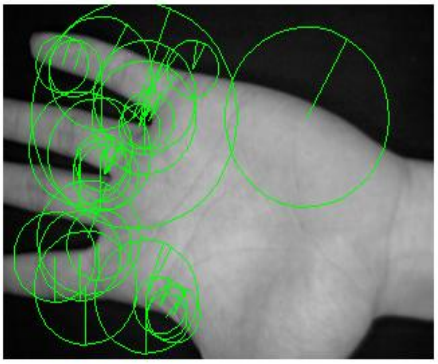

Figure 5. (a) Keypoints shown on palm image of subject 1

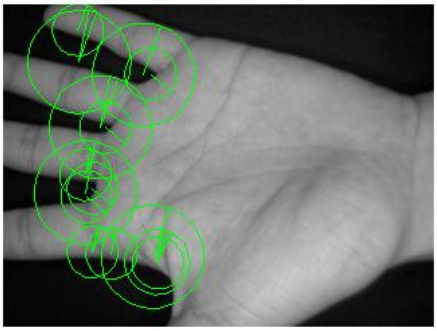

Figure 5. (b) Keypoints shown on palm image of subject 2

### *B. Results of using SIFT on palm for matching under controlled environment*

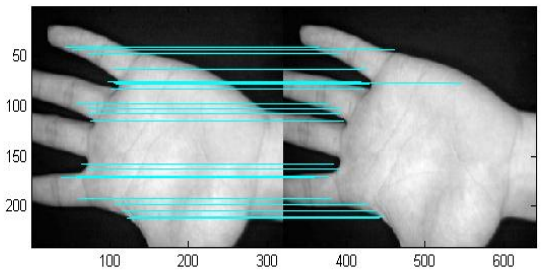


Figure 6. (a) Keypoint matching of same palm of subject 1 (37 matches found out of 37)

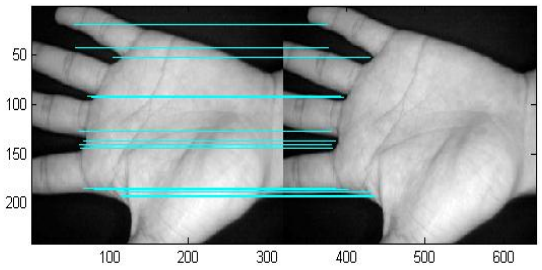


Figure 6. (b) Keypoint matching of same palm of subject 2 (23 matches found out of 23)

### *C. Results of usingSIFT on palm for matching after rotation*

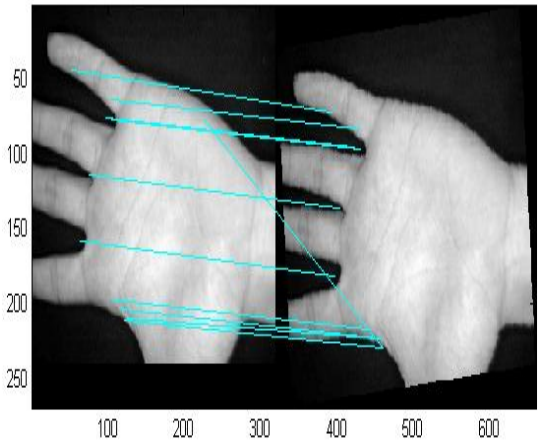


Figure 7. (a) Keypoint matching after rotation by 5 degrees in counterclockwise direction (13 matches found out of 37)

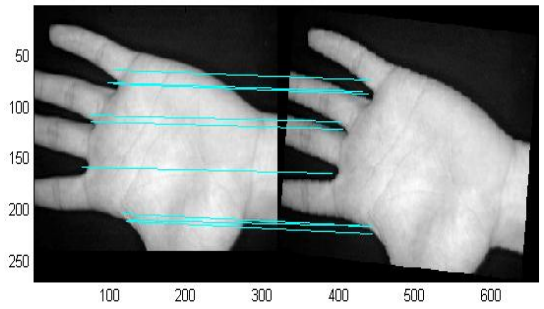

Figure 7. (b) Keypoint matching after rotation by 5 degrees in clockwise direction (10 matches found out of 37)

### *D. Result of using SIFT on palm for matching after changing the scale*

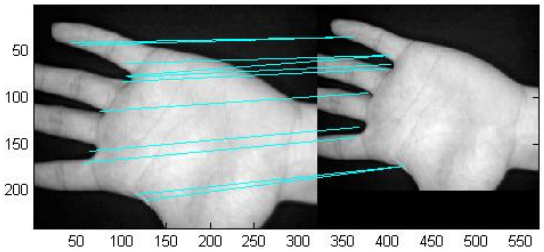

Figure 8. Keypoint matching after changing the scale (18 matches found out of 37)

### *E. Results of using SIFT on palm for matching after changing illumination*

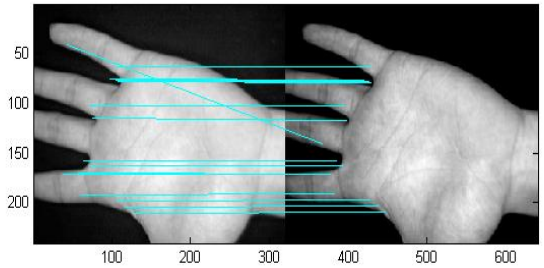

Figure 9. (a) Keypoint matching in dark environment (18 matches found out of 37)

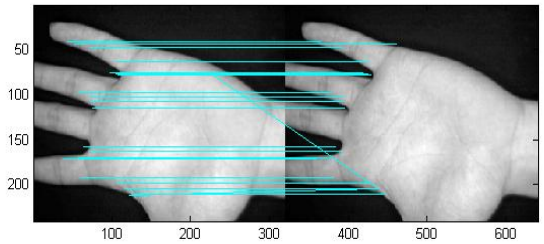

Figure 9. (b) Keypoint matching in bright environment (31 matches found out of 37)

### *F. Result of using SIFT on palm after changing contrast*

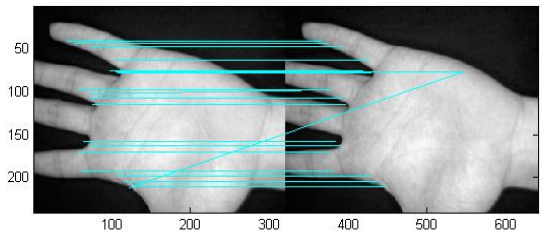

Figure 10. Keypoint matching after changing contrast (27 matches found out of 37)

### *G. Results of using SIFT for matching palms of different subjects*

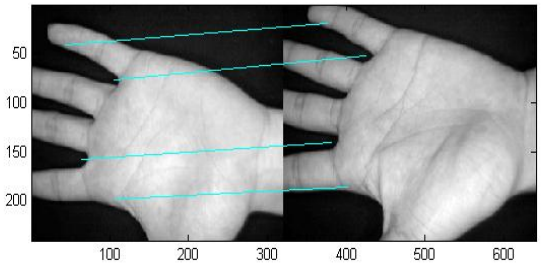

Figure 11. (a) Keypoint matching of palm of different subjects (only 4 matches found)

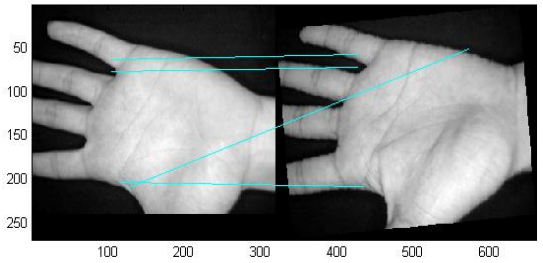

Figure 11. (b) Keypoint matching of different subjects after rotation in counterclockwise direction (only 4 matches found)

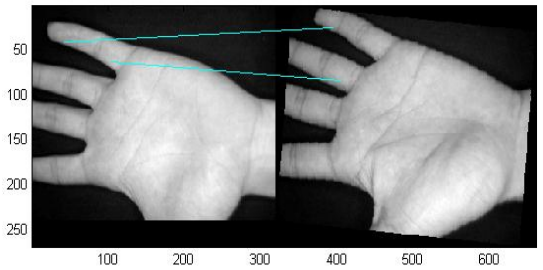

Figure 11. (c) Keypoint matching of different subjects after rotation in clockwise direction (only 2 matches found)

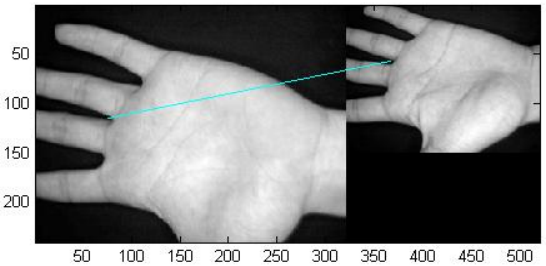

Figure 11. (d) Keypoint matching of different subjects after changing the scale (only 1 match found)

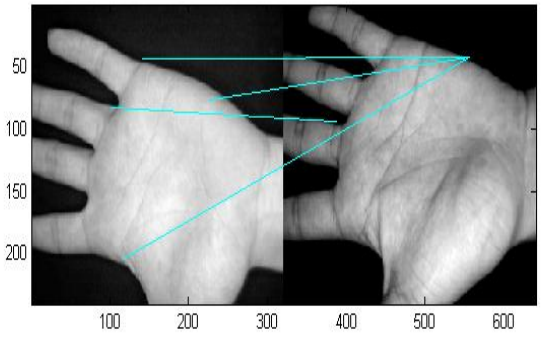

Figure 11. (e) Keypoint matching of different subjects in dark environment (only 4 matches found)

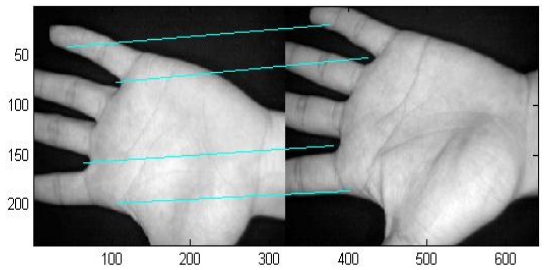


Figure 11. (f) Keypoint matching of different subjects in bright environment (only 4 matches found)

## V. CONCLUSION

From several experiments conducted we conclude that the SIFT algorithm gives good results for matching under both controlled and strained environment. It is robust to changes in scale and rotation. It also gives good results for small to medium illumination changes but not for large changes. It can be used for small occlusion but not for large occlusion. It is an efficient algorithm for recognition and detection purpose for small database but not so good (with respect to computational speed) for large database.

## VI. FUTURE SCOPE

In SIFT edges are poorly defined and usually hard to detect, but there are still large numbers of keypoints can be extracted from typical images. Also sometimes the images are too smooth to find that many SIFT features for a matching, and in that case a small image could be unrecognized from the training images. The recognition performance could be improved by adding new SIFT features or varying feature sizes and offsets.

## REFERENCES

[1] D. Zhang, W. K. Kong, J. You, and M. Wong, "Online palmprint identification," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, no. 9, pp. 1041-1050, September 2003.
[2] D.D. Zhang, Palmprint Authentication. Norwell, Mass. Kluwer Academic Publishers, 2004.
[3] D. Maltoni, D. Maio, A.K. Jain, and S. Prabhakar, Handbook of Fingerprint Recognition, Springer, 2003.
[4] A.K. Jain, A. Ross, "A Prototype Hand Geometry based Verification System," proc. of AVBPA, 166-171, 1999.
[5] David G. Lowe, "Distinctive Image Feature from Scale-Invariants Keypoints," International Journal of Computer Vision, vol. 60 (2), pp. 91-110, 2004.
[6] Cong Geng, Xudond Jiang, "Face Recognition Using SIFT Feature," ICIP, IEEE, 2001.
[7] U. Park, S. Pankanti, A.K. Jain, "Fingerprint Verification Using SIFT Feature," proc. of SPIE6944, 2008.
[8] David G. Lowe, "Object Recognition from Local Scale-Invariant Features," The Proceedings of the Seventh IEEE International Conference on Computer Vision, vol.2, pp.1150-1157, 1999.
[9] C. Zhang, Zhihui Gong, Lei Sun, "Improved SIFT Feature applied in image Matching," Computer Engineering and Application, vol.2, pp.95-97, 2008.
[10] David G. Lowe, " Local feature view clustering for 3D object recognition," IEEE Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, pp. 682-688, 2001.

**Himanshu Maurya** received the B.E. degree in Electronics and Communication Engineering from Thakral college of Technology in 2011, and presently pursuing M.Tech in Communication System from National Institute of Technology, Patna, India

**Shikha Maurya** received the B.Tech degree in Electronics and Communication Engineering from B. N. College of Engineering of Technology in 2012, and presently pursuing M.Tech in digital Communication from ABVIIITM, Gwalior, India.

**Prof. Bikas Chandra Sahana** received the Ph.D. degree from National Institute of Technology, Patna, India, and presently working as Assistant Prof. in Electronics & Communication Engineering department in NIT, Patna.