International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2022): 7.942

Feature Extraction Algorithms for Biometric Authentication using Palm Veins

Ahmed S. Gheith, Hossam Eldin Mostafa

Electronics & Communications Department, Faculty of Engineering, Masnoura University, Egypt

Abstract: ***This paper is inspired by the importance of feature extraction algorithms for biometric applications, a robust way to detect features and extract them, that can be used later for matching and hence recognition and verification, in this paper we will compare various techniques SIFT, ORB, SURF in terms of feature extraction performance, feature extraction quality and matching performance for biometric palm vein samples.***

Keywords: SIFT SURF ORB Biometric Palm veins

1. Introduction

Image features are any piece of information that can help identify/solve a computational task for a computer vision application, features have many types (edges, corners, blobs, ridges) and in most of computer vision applications features are constructed from a mix of the previously mentioned types, then we construct a features vector for each feature point this is commonly called a feature vector, and combining the set of all feature vectors in an image is called feature space.

The first algorithm we will explore is Scale Invariant Feature Transform (SIFT), SIFT is a feature detector developed by Lowe in 2004 [1]. SIFT has widely used object recognition applications, however, the drawback is that it requires extensive computational power, especially for real-time applications.

The second algorithm that we will explore is Speed up Robust Feature (SURF), SURF is a variant of SIFT that approximate its algorithms, It has better performance than SIFT and consumes less computational power without sacrificing the quality of the features. Both SIFT and SURF are using algorithms that is based on the same concepts of a descriptor and a detector.

The third algorithm that we will explore is ORB which is a fusion between two algorithms Binary Robust Independent Elementary Features (BRIEF) which is a SIFT alternative that needs less computational power than SIFT with good and comparable matching performance and FAST keypoint detector.

In this paper, we will conduct a performance comparison of the three mentioned algorithms SIFT, SURF, and ORB in the field of Biometrics, we will focus on the quality of the detected features/ key points and the accuracy of matching.

The sample data that we will use is a plam veins image that is captured using a special camera with IR filters on 850~950 nm.

2. Literature Review

SIFT

Scale-invariant feature transform (SIFT) [2]is an algorithm that solves multiple issues with features detection like intensity, image rotation, viewpoint change, and affine transformations. The SIFT algorithm has four main stages. The first stage is to find scale-space extrema using the Difference of Gaussian (DoG) algorithm. The second stage is key point localization in which we eliminate the low contrast points and get refined and localized points. The third stage is using the local image gradient to assign an orientation to the key points. The fourth stage is called descriptor generator by using each image gradient magnitude and orientation to calculate a local image descriptor for all key points.

"RootSIFT" a scale and rotation invariant feature extractor; a modified version of SIFT using Hellinger kernel instead of Euclidean distance.

SIFT algorithm is mainly constructed from the following stages:

(1) Detection of Scale-space extrema

To calculate the scale-space we begin by detecting the keypoints. A convolution of Gaussian filters are applied to the image at different scales, next we calculate the difference of successive Gaussian-blurred image. Then key points are calculated form the maximum and minimum of the Difference of Gaussians (DoG) that was calculated at different scales. Specifically, a DoG image $D(x, y, \sigma)$ is given by:

$$D(x, y, \sigma) = L(x, y, ki\,\sigma) - L(x, y, kj\,\sigma)$$

Where $L(x, y, k\sigma)$ is the convolution of the original image $I(x, y)$ with the Gaussian blur $G(x, y, k\sigma)$ at scale $k\sigma$, $L(x, y, k\sigma) = G(x, y, k\sigma) * I(x, y)$

(2) key-point localization

The previous calculation of scale-space extrema produces a lot of candidate key points; some of them are inaccurate and unstable. To find accurate key point parameters, the next step in the algorithm is to use the nearby data to perform a detailed fit and produce an accurate scale, location, and ratio

of the primary curvatures. According to the provided information, the key points with the lowest contrast which means they will be susceptible to noise or are not correctly localized and near to an edge are refused.

- Interpolation of neighboring data for precise position
- Rejecting key points with the lowest contrast
- Rejecting edge features responses

(3) Keypoints orientation assignment

In this step, every of the key points is assigned to a single or multiple orientations in regards to the directions of the gradient of the local image. This is the main step to have key points that are rotationally invariant, as the key point's descriptor can be calculated in correspondence to its orientation so that we achieve the desired rotation invariance.

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

$$\theta(x,y) = \text{atan2}(L(x,y+1) - L(x,y-1), L(x+1,y) - L(x-1,y))$$

We achieve that by calculating the direction and magnitude of the gradient in the Gaussian-blurred image for all pixels in a neighborhood of pixels around the key point. Then we form a histogram with 36 bins for the orientation, Each of those bins is covering 10 degrees and thus covering all 360 degrees of orientation,. Then all samples in the neighborhood window are added to a histogram bin and are weighted by its magnitude of gradient and by a weighted gaussian circular window with a σ that is 1.5 times that of the scale of the key point. Then we calculate the dominant orientation based on the peaks in this histogram. Once we fill the histogram calculations, We assign to the key points the orientations corresponding to the highest and local peaks that are in the range of 80% of the highest peaks. If we find multiple orientations, we can create an extra key point with the same scale and location of the original key point for every found orientation.

(4) generation of key-point descriptors

In the previous stages we have calculated the locations of the key points at defined scales and assigned the calculated orientations to each of the key points. This achieved location invariance, scale invariance, and rotation in variance. In this stage, a descriptor vector is computed for every key point on the image most similar in scale to the scale of the key point, such that the descriptor is partially invariant and highly distinctive to the other variations like changes in illumination, field of view, etc.

First, we create a set of orientation histograms on 4×4 pixel neighborhoods with each having 8 bins. Then we use 16×16 neighboring regions around the key point in order to have each histogram constructed of samples from a 4×4 sub region of the original neighborhood area to compute these set of orientation histograms from magnitude and orientation values.

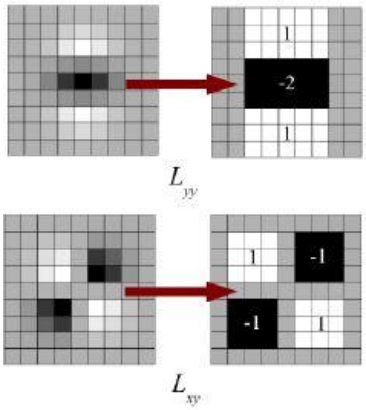
Then the magnitudes are weighted by a Gaussian function with a σ equal to one half the width of the descriptor window, which results a vector of all of the values of these histograms. The resulted vector has 128 elements, since there are 4 × 4 = 16 histograms each with 8 bins. Then we normalize this vector to unit length that enhances invariance to changes in illumination.

(5) Representing image features by the formation of several descriptors of 128-dimension.

SURF

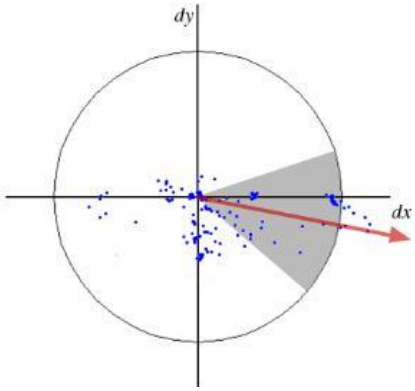
Speeded up robust features (SURF) is inspired by the need of having a local feature detector and descriptor that is faster then SIFT. The authors of SURF claims that it is several times faster than SIFT and they claim it to be more invariant to different image transformations. It can be used for tasks such as image registration, image classification, object recognition, and 3D image reconstruction.

In SIFT, to find scale-space the algorithm approximated Laplacian of Gaussian with Difference of Gaussian. SURF adds to the SIFT algorithm and approximates LoG with Box Filter. Such an approximation is shown at the below image. The convolution of box filters can be done in a less computationally demanding way using the help of integral image which is a big advantage of this approximation. And since we can execute this approach in parallel for different scales, which maximizes the advantage even more. SURF calculates scale and location using the determinant of Hessian matrix.



SURF assigns the orientation of the features by using the responses of the wavelet in both of the horizontal and the vertical directions for a neighboring region of size 6s and applies proper gaussian weights to it. Then we plot the result in a space as shown in the shown image. Then we estimate the main orientation by computing the sum of all the responses within an angle of 60 degrees of a sliding orientation window. The interesting part is, we can calculate the wavelet responses using integral images with an ease at any scale. And since not all applications require invariance to rotation, SURF doesn't calculate the feature orientation, which speeds up the calculation time. SURF provides an orientation calculation functionality called Upright-SURF or

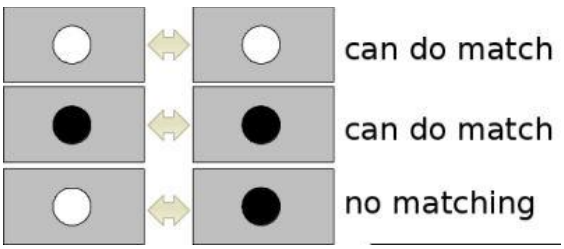
U-SURF. It improves its computational complexity and is potency to change by up to 15 degrees.



For the feature descriptors, SURF uses the responses of the wavelets in both of the vertical and the horizontal directions with the use of integral images. The calculation is done by taking a neighboring region of size 20sX20s is around the key point where s is the original size. Then it is divided into 4x4 sub regions. Then a vector$v = (\sum dx, \sum dy, \sum|dx|, \sum|dy|)$ is formed from each of them, a horizontal wavelet and vertical wavelet responses. Accordingly the calculated feature descriptor has a total of 64 dimensions. We can achieve higher computational and matching speeds by lowering the dimensions, but in sacrificing the distinctiveness of the features.

SURF feature descriptor can be more distinctive by extending dimensions to 128. In this version the sums of dx and |dx| are calculated independently for dy<0 and dy≥0. Similar to them, the sums of dy and |dy| are separated according to the sign of dx, which doubles the total number of the computed features. And more significantly this approach is not adding much to the complexity of the algorithm and accordingly doesn't add a lot of the needed computational resources.

Another one significant improvement in SURF is the use of the sign of Laplacian (trace of Hessian Matrix) for the underlying interest points. It adds as little as zero computation complexity because we compute it during the same detection phase. The sign of the Laplacian helps distinguish bright areas on darker backgrounds from the opposite of darker areas on bright backgrounds. At the matching step, we will only consider features for comparison if they have a similar contrast type. This simple facts enhances the matching speed and hence the whole matching speed, without sacrificing the performance of the descriptor's.



ORB

ORB is an efficient alternative to SIFT or SURF.ORB is its core is a merge of FAST key point detector and BRIEF descriptor with a lot of modifications that enhances the performance. First ORB uses the FAST algorithm to calculate the key points, next it uses Harris corner algorithm to find the top N points. And also produces multiscale features by using pyramids. The disadvantage is that FAST doesn't calculate the orientation. But ORB comes up with a modification that solves the orientation disadvantage.

The algorithm calculates the intensity weighted centroid of the patch with a located corner at the center. Then finding the orientation through the calculated vector's direction from the located corner point to the center. To enhance the invariance of rotation, the algorithm computes the moments with x and y in a circular region of radius r, where r is the size of the patch.

For calculating the descriptors, ORB uses BRIEF descriptors. However BRIEF has the same disadvantage as rotation, So to overcome the issue ORB "steer" BRIEF relative to the orientation of keypoints. We define 2×n matrix S for any feature set of n binary tests at location (xi,yi), which contains the coordinates of these pixels. Then we use the orientation of the patch, θ, to find its rotation matrix and then rotates the S to get steered(rotated) version Sθ.

ORB discretizes the angle to increments of twelve degrees (2π/30), then we precompute BRIEF patterns and build a lookup table of them. As long as we have a consistent keypoint orientation θ across views, then we calculate the descriptor by using the correct set of its points Sθ.

One of the most important properties of BRIEF is that each one of the features has a mean near 0.5 and a large variance. But when it has an orientation along a keypoint direction, it becomes separated and loses this property. Having a feature with high variance makes it more discriminative because it will respond differently to different inputs. Also another important feature is to have uncorrelated tests, so that each test will have a specific contribution to the result. To resolve what we have just illustrated, ORB uses a greedy search algorithm for all viable binary tests to find the tests that will have both a means close to 0.5 and a high value of variance and, and also uncorrelated. The result is called of the above algorithm is called **rBRIEF**.

At the descriptor matching stage, ORB uses multi-probe LSH which is better than the normal LSH. The authors of claim that ORB is less computationally extensive than SURF and SIFT and that ORB descriptors are better than SURF.

Features Extraction Comparison

In this section we researched and compared the performance and the quality of the key points for each of the three methods (SIFT, SURF and ORB) in palm veins samples. The results are indicated at Figures 1 for SIFT, SURF and ORB respectively which indicates a better key points quality for SIFT and SURF compared to ORB, however ORB is much faster to calculate as shown in Table1.

Table 1: Key points performance comparison

	Time (sec)	Key points
SIFT	0.0642	363
SURF	0.2858	308
ORB	0.0034	580

Figure 1: SIFT, SURF and ORB keypoints quality ordered from the left

Features Matching Comparison

In this section we researched and compared the performance and match rate for each of the three methods (SIFT, SURF and ORB) against each of the scale, skew and rotation changes.

Scale changes

We applied a scaling of 10% of the original image here, and the results at figures 2, 3, 4 show ORB to be the most sensitive and weakest among the other algorithms.

Skew changes

We also applied a 10% skew change to the image changing the perspectives and the field of view, and we have noticed the same results that shows ORB as the worst performer while SIFT is the best among the three algorithms.

Rotation changes

The final change was applying a rotation by 45 degrees, and as shown at the results at Table-2 SIFT was the best performer by a matching rate of 56.25% followed by SURF with a matching rate of 33.66% followed by ORB by 17.77%.

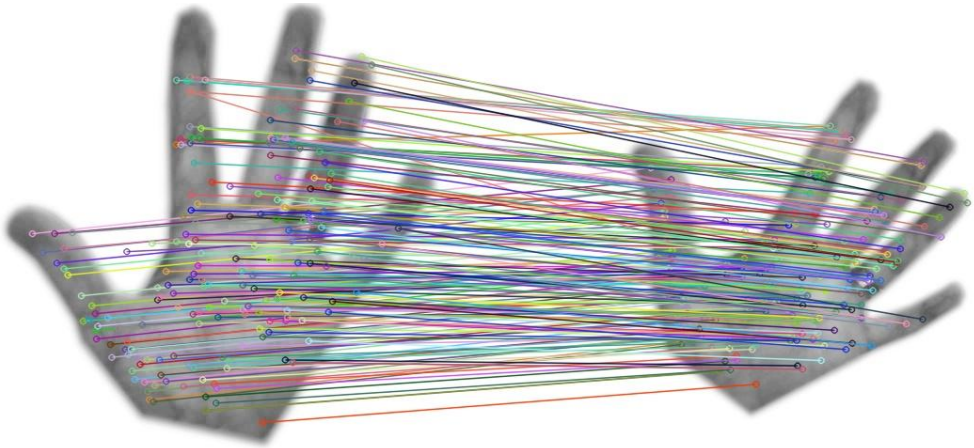


Figure 2: SIFT matching

Figure 3: SURF matching

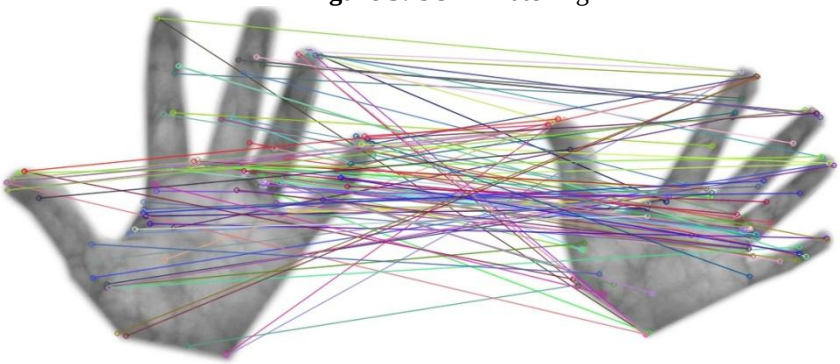


Figure 4: ORB matching

Matching results are calculated based on the average of number of keypoints compared to the number of matches with a brute-force distance less than 0.75, results of the matching are illustrated at Table 2.

Table 2: Matching performance comparison

	Time (sec)	Keypoints- 1	Keypoints- 2	No. of matches	Match rate
SIFT	0.1510	363	341	198	56.25%
SURF	0.7123	308	298	102	33.66%
ORB	0.0127	580	681	112	17.77%

3. Conclusion

We compared in this paper different matching techniques and algorithms for the purpose of biometric authentication and significantly palm veins features and we applied during our testing and experimental a 10% scale, a 10% skew and a 45-degree rotation, and we displayed the quality of the key points, performance of the algorithm in terms of needed computational power and matching rate of the distorted images of all of the three techniques (SIFT, SURF, and ORB), the results showed that ORB has the fastest computational performance, however it has the worst quality of key points and the worst matching rates, the results also showed that SURF has the worst computational performance with a mediocre key points quality and mediocre match rates, and finally the results showed that SIFT is the best performer in terms of palm veins features extraction with good key points quality, good computational performance and higher match rates.

References

[1] David G Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” *International Journal of Computer Vision*, vol.50, No. 2, 2004, pp.91-110.

[2] E. Karami, M. Shehata, A. Smith, “Image Identification Using SIFT Algorithm: Performance Analysis Against Different Image Deformations,” in Proceedings of the 2015 Newfoundland Electrical and Computer Engineering Conference, St. john’s, Canada, November, 2015.

[3] Liang-Chi Chiu, Tian-Sheuan Chang, Jiun-Yen Chen and N. Chang, 'Fast SIFT Design for Real-Time Visual Feature Extraction', *IEEE Trans. on Image Process.*, vol. 22, no. 8, pp. 3158-3167, 2013.

[4] Y. Ke and R. Sukthankar, “PCA-SIFT: A more distinctive representation for local image descriptors.” in *Proc. CVPR.* Vol. 2, pp. 506–513, 2004.

[5] Herbert Bay, Tinne Tuytelaars and Luc Van Gool, "Speeded-up robust features (SURF)," *Computer vision and image understanding*, vol.110, No.3, 2008, pp. 346-359.

[6] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. “BRIEF: Binary robust independent elementary features,” *In European Conference on Computer Vision*, 2010.

[7] Ethan Rublee, Vincent Rabaud, Kurt Konolige and Gary Bradski, “ORB: and efficient alternative to SIFT or SURF,” *IEEE International Conference on Computer Vision*, 2011.

[8] P. Sykora, P. Kamencay and R. Hudec, “Comparison of SIFT and SURF Methods for Use on Hand Gesture Recognition based on Depth Map”, *AASRI Procedia*, vol. 9, pp. 19-24, 2014.

[9] P. M. Panchal, S. R. Panchal, and S. K. Shah, "A Comparison of SIFT and SURF," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 1, no. 2, pp. 323-327, 2013.
[10] N. Y. Khan, B. McCane, G. Wyvill, "SIFT and SURF Performance Evaluation Against Various Image Deformations on Benchmark Dataset," in *Proceedings of 2011 International Conference of Digital Image Computing Techniques and Applications*.
[11] B. Moghaddam, C. Nastar and A. Pentland, "A Bayesian similarity measure for deformable image matching," *Image and Vision Computing*, vol. 19, no. 5, pp. 235-244, 2001.
[12] B. Shan, "A Novel Image Correlation Matching Approach," *JMM*, vol. 5, no. 3, 2010.
[13] M. Güzel, 'A Hybrid Feature Extractor using Fast Hessian Detector and SIFT', *Technologies*, vol. 3, no. 2, pp. 103-110, 2015.
[14] Zhen Liu, Ziying Zhao, Yida Fan, Dong Tian, **"**Automated change detection of multi-level icebergs near Mertz Glacier region using feature vector matching," *The international conference on Image processing, computer vision and Pattern Recogniton*, 2013.
[15] R. Kwok and N. Untersteiner, "The thinning of Arctic sea ice," *Physics Today*, vol. 64, no. 4, p. 36, 2011.
[16] J. Faidit, "Planetariums in the world," *Proceedings of the International Astronomical Union*, vol. 5, no. 260, 2009.