HOOFR: An Enhanced Bio-Inspired Feature Extractor

Dai-Duong Nguyen, Abdelhafid El Ouardi, Emanuel Aldea, Samir Bouaziz
SATIE - CNRS UMR 8029
Paris-Sud University, Paris-Saclay University, France
{dai-duong.nguyen, abdelhafid.elouardi, emanuel.aldea, samir.bouaziz}@u-psud.fr

***Abstract*—Feature matching plays an important role in many computer vision applications, such as object recognition, scene reconstruction or image mosaicing. In this paper, we propose an algorithm called Hessian ORB - Overlapped FREAK (HOOFR) which is based on the combination of the ORB detector and the FREAK bio-inspired descriptor. We address some modifications related to the detection and the description processes in order to enhance HOOFR reliability, speed and memory fingerprint. The experiments on a widely used dataset demonstrate the considerable performance of HOOFR compared to SIFT, SURF or ORB in terms of the execution time and the matching quality, in various matching contexts.**

***Index Terms*—SIFT, SURF, ORB, FREAK, Hessian matrix, Feature matching.**

I. Introduction

Feature matching is the task of establishing the correspondences between two images of the same scene and many vision applications rely on the stability of the matching result. Through over a decade old, the most popular algorithm is Scale Invariance Feature Transform (SIFT) proposed by Lowe [1]. SIFT identifies keypoints based on the local extremum of Different of Gaussian (DoG) over scale space and describes them by a 3D spectral histogram of the image gradients. SIFT is remarkably successful in object recognition [1], visual mapping [2], automatic parorama[3], etc. However, it is affected by high computation requirements, which prohibit its implementation in real-time applications such as visual odometry, or on low-power embedded devices such as mobile phones. An alternative named Speed Up Robust Feature (SURF) was proposed in [4]. This method relies on the determinant of the Hessian matrix for keypoint detection and on the responses of Haar-like filters for the description. SURF has a comparable performance to SIFT but it exhibits a significant improvement in computation speed. The reason is that while SIFT approximates Laplacian of Gaussian (LoG) by DoG, SURF goes further and approximates LoG by box filters. By relying on an integral image, the box filter convolution may be performed efficiently. Then, two sets of SIFT or SURF keypoints may be matched by employing Euclidean floating distances among descriptors.

On the other end of the spectrum, to address real-time applications, ORB [5] uses a binary representation in order to simplify the calculation. ORB is inspired by the FAST [6] keypoint detector and by the BRIEF [7] descriptor. In fact, FAST does not provide neither multi-scale features nor orientation measurement. Therefore, in ORB the authors employs a scale pyramid representation and detect FAST features at each level; additionally, keypoint orientation is estimated using the local intensity centroid. The ORB descriptor is then constructed based on rotated BRIEF which uses simple binary tests between pixels in a smoothed image patch. ORB algorithm offers a high efficiency to be implemented in patch-tracking application on smart phone [5], image matching on Android devices [8] or SLAM application [9], etc.

Apart from BRIEF, there are several other variants of binary descriptors, among which BRISK[10] and FREAK [11]. A clear advantage of binary descriptors is that the Hamming binary distance may replace the Euclidean floating distance for matching, by using bit-wise XOR followed by a bit count on specific architectures, which is significantly faster. The key concept of the BRISK descriptor is the use of a symmetrical pattern. Instead of random points as in BRIEF, sampling points of BRISK are located on circles concentric to the keypoint. Furthermore, BRISK divides sampling-point pairs into two subsets: *long-distance pairs* reserved to compute keypoint orientation and *short-distance pairs* reserved to build keypoint descriptor. Following this idea, FREAK is an optimized version of BRISK with two main modifications. Firstly, it uses a sampling pattern inspired from the human retina where the smoothing kernels are overlapping and their size exhibit exponential change. Secondly, it uses 45 symmetrical pairs with respect to the center to estimate keypoint orientation rather than using the *long-distance pair* subset as in BRISK.

Our proposed feature is an optimized combination of the ORB detector and FREAK descriptor. In order to filter keypoints at each level of the scale pyramid more accurately, we replace the Harris response in ORB by the Hessian response. For the description, we modify the sampling pattern of FREAK in order to have more overlapping between smoothed kernels, which allows us to reduce the number of pairs in calculating keypoint orientation. We refer to this combination as HOOFR and will detail the modifications in section II. Afterward, to validate the performance of HOOFR, we will present in section III the experimental tests on a real benchmark dataset.

II. HOOFR

A. FAST detection filtered by Hessian matrix

FAST [6] detector is widely used due to its computational properties. It considers the points on a circular ring around

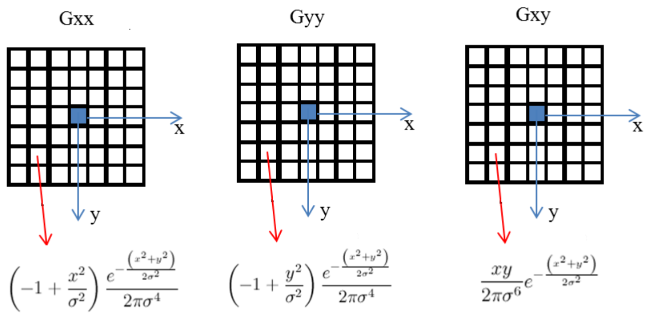

Gxx
Gyy
Gxy
x
y
$\left(-1+\frac{x^2}{\sigma^2}\right)\frac{e^{-\frac{(x^2+y^2)}{2\sigma^2}}}{2\pi\sigma^4}$
$\left(-1+\frac{y^2}{\sigma^2}\right)\frac{e^{-\frac{(x^2+y^2)}{2\sigma^2}}}{2\pi\sigma^4}$
$\frac{xy}{2\pi\sigma^6}e^{-\frac{(x^2+y^2)}{2\sigma^2}}$

Fig. 1. Square filters for calculating the Hessian matrix in HOOFR

one pixel. In case of enough consecutive pixels on the ring which are brighter or darker than the central pixel with a threshold t, this latter pixel is considered as a corner. The number of consecutive pixels is generally set between 9 and 12 depending on the application. FAST-9 is employed in ORB to detect features at all level of the scale pyramid. Each level is an image sampled from original image at one corresponding scale. Due to the fact that FAST provides a significant number of features and it has a large response along edges, the Harris matrix was used to filter the result.

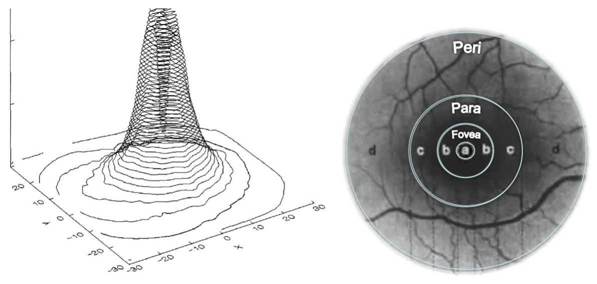
In HOOFR, we apply a similar detection method to ORB. However, we are inspired by the overall results of [12] who evaluated different detection methods based on Harris, Hessian or MSER. These results show that in general, the Hessian based detection overcomes that based on Harris. Hence, instead of the Harris matrix, we propose to employ the Hessian matrix illustrated in equation 1 in order to filter the features provided by FAST.

$$H = \begin{bmatrix} \frac{\partial^2 I}{\partial x^2} & \frac{\partial^2 I}{\partial x \partial y} \\ \frac{\partial^2 I}{\partial x \partial y} & \frac{\partial^2 I}{\partial y^2} \end{bmatrix} \tag{1}$$

The Hessian matrix consists of the second order partial derivatives of the image. The eigenvectors of this matrix form an orthogonal basis highlighting the local direction of the gradient. If the product of eigenvalues of the Hessian matrix is positive, a local extremum is present. We note that for any square matrix, the product of eigenvalues is the determinant of the matrix. Another detector relying on this determinant with remarkable results is SURF[4]; therefore, in HOOFR, we propose to use the determinant of the Hessian matrix as the score of the feature point.

In general, in order to find the derivative, the image is first smoothed and then the numerical approximations are applied as this operation is sensitive to noise. Nevertheless, instead of employing an averaging filter to smooth the image and then finding its derivative, the derivative can be directly applied to the smoothing function which can then be used to filter the image. This would also make it computationally efficient. In HOOFR, we use Gaussian shown in equation 2 as a smoothing function.

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



(a) Density of ganglion cells over the retina (b) Retina areas

Fig. 2. Distribution of ganglion cells over the retina [11]. There are four areas of the density: (a) foveal, (b) fovea, (c) parafoveal and (d) perifoveal

For each candidate point returned by FAST, we calculate its Hessian matrix. In practice, each element of this matrix is generated by applying a square filter with the dimension of 7x7 shown in figure 1 corresponding to the second order derivative of the smoothing function. Then, the determinant of this matrix is considered as the score of the point. If there are more than K points detected by FAST, we only maintain the K points which exhibit the highest score.

B. *FREAK - bio-inspired descriptor*

FREAK was proposed by [11] by considering human retina topology and neuroscience observations. It is believed that human retina extracts information from the visual field by using the Gaussian comparison (Difference of Gaussian) of various sizes and by encoding these differences in binary mode as a neural network.

1) Sampling pattern: The topology and spatial encoding of the retina is interesting. First, a ganglion cell includes several photoreceptors. The region where light influences the response of a ganglion cell is the receptive field. Figure 2 shows that the spatial distribution of ganglion cells reduces exponentially with the distance to the foveal. They are segmented into four areas: foveal, fovea, parafoveal, and perifoveal. Furthermore, the sizes of the receptive field and dendritic field increase with the radial distance to the foveal.

Inspired by this idea, the authors of [11] proposed a sampling pattern as showed in Figure 3a. The pattern is composed of 7 concentric circles with exponentially decreasing radius. Each circle contains 6 points considered as 6 receptive fields, and the receptive field at the center, so that the overall pattern is formed by 43 receptive fields. The distribution of the points on the concentric circles is similar to the method of 6-segments presented in DAISY [13].

With HOOFR, we propose a different sampling pattern illustrated in figure 3b. Our sampling pattern contains only 6 concentric circles. However, each circle has 8 receptive fields distributed as the 8-segment method in DAISY. Therefore, including the point at the center, this pattern contains 49 receptive fields in total. The justification for our proposed configuration is that for complex image processing tasks, various

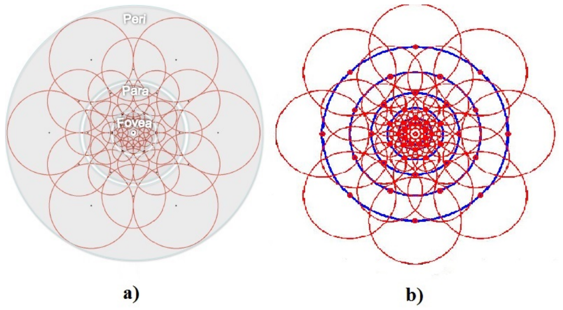


Fig. 3. Sampling pattern in FREAK [11] (a) and in HOOFR(b)

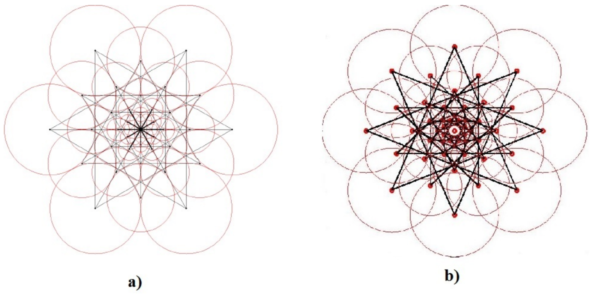


Fig. 4. Illustration of selected pairs to estimate the orientation in FREAK [11] (a) and HOOFR(b)

descriptors exploit, either in the image space [14] or in the frequency domain[15], a certain degree of overlapping in order to be able to grasp more effectively complex correlations. With respect to FREAK, our configuration increases, in addition to the radial overlap, the amount of circumferential overlap among the fields.

Due to the fact that FREAK uses the comparison between these receptive fields to build the descriptor, with 49 fields, we have more pairs (1176 pairs) to choose than that of [11] (903 pairs). Moreover, in our sampling pattern, we have the overlap not only between the receptive fields of different concentric circles but also circumferentially.

2) Keypoint orientation: In order to estimate the keypoint orientation, we use the same method proposed in FREAK by summing the local gradients over selected pairs. However, our sampling pattern has more overlapping leading to more information being integrated in the receptive field. Hence, we can use fewer pairs than FREAK for orientation estimation. The latter is using 45 pairs with symmetric receptive fields with respect to the center as shown in figure 4a, whereas we select only 40 pairs as shown in figure 4b. By decreasing the number of pairs, we can improve the execution time when computing the orientation.

The orientation is then obtained by the equation 3 where S is the set of all 40 pairs used to compute local gradients, N is number of pairs in S and P_0^{r1} is the 2D vector of coordinates of the receptive field center. The space of orientation in HOOFR is also discretized by the same steps proposed in FREAK.

$$O = \frac{1}{N} \sum_{P_0 \epsilon S} (I(P_0^{r_1}) - I(P_0^{r_2})) \frac{P_0^{r_1} - P_0^{r_2}}{\| P_0^{r_1} - P_0^{r_2} \|} \tag{3}$$

3) Descriptor: The binary descriptor F is constructed by the comparison between receptive fields with their corresponding Gaussian kernel.

$$F = \sum_{0 \leq n < N} 2^n T(P_n) \tag{4}$$

$$T(P_n) = \left\{ \begin{array}{lll} 1 & if & (I(P_n^{r_1}) - I(P_n^{r_2})) > 0 \\ 0 & otherwise & \end{array} \right. \tag{5}$$

where P_n is the pair of receptive fields, N the size of binary descriptor, $I(P_n^{r_1})$ and $I(P_n^{r_2})$ are respectively the Gaussian smoothed intensities of the first and the second receptive field of the pair n.

Here, we experience a second advantage of the increase in overlap, the fact that it contributes to reducing the descriptor size. In HOOFR, we build a descriptor of size 256 bits which is half the size of the FREAK descriptor (512 bits). This reduction is aimed not only at memory-saving, but also at accelerating the matching process where the 256-bits comparison is two times faster than 512-bits comparison. In fact, following testing, we found that a 256-bits descriptor is high enough to ensure a good performance for our sampling pattern. This boils down to selecting the 256 most relevant pairs among the total of 1176 pairs. These pairs are also chosen experimentally by running an algorithm similar to the ORB selection. This algorithm has 3 main steps:

- The first step extracts keypoints from training data. We take all the possible pairs (1176 pairs) to build the description and each keypoint has its own descriptor. A matrix M is created where the number of rows corresponds to the number of keypoints and the number of columns corresponds to the size of descriptor (1176 columns).
- For each column, we calculate the average which is situated between 0 and 1. This value represents the variance of the binary distribution. The high variance is desired to have a discriminant feature and the mean of 0.5 leads to the highest variance.
- All the columns are ordered and we keep the 256 columns which have the highest variances.

Figure 5 shows the 256 relevant selected pairs used in HOOFR.

III. Performance Evaluation

Our proposed algorithm has been tested using the well-known evaluation method and datasets published by Mikolajczyk and Schmid[16]. We take eight image sequences as shown in figure 6, corresponding to viewpoint change (Graffiti, Wall), zoom and rotation (Bark, Boat), blur (Bikes, Trees), brightness change (Cars) and JPEG compression (Ubc) to evaluate the performances.

Each sequence contains 6 images ordered by the increasing amount of transformation from image 1 to image 6. All

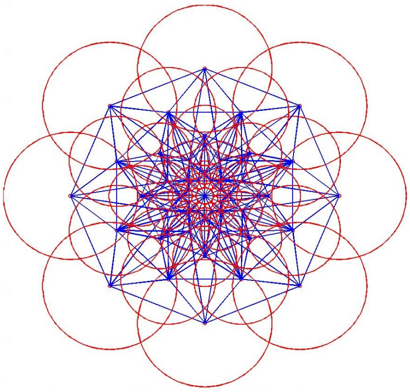


Fig. 5. Illustration of 256 selected pairs used to construct the descriptor in HOOFR

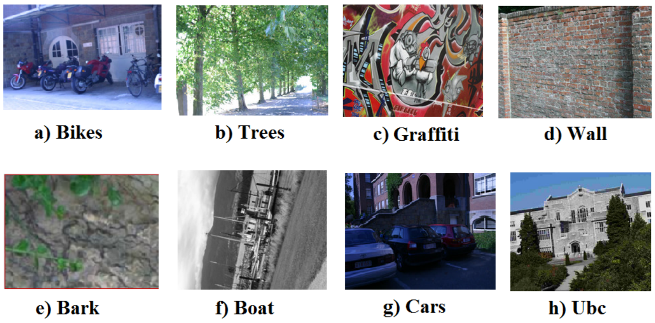


Fig. 6. Image sequences used for evaluation

transformations are planar, so ground truth is determined based on the homography matrix. Furthermore, matching is performed between each image and the first image of the same sequence because homography matrices for these pairs of images are carefully defined in the datasets. We consider that a point p_a in one image is a correspondence of a point p_b in other image when they satisfy two conditions:

- The error in relative location of $||p_a - H \cdot p_b|| < 1.5$ pixel where H is the homography matrix between the two images.
- The overlap area of the keypoint region in one image and the projection of the keypoint region from the other image is high enough. In our test, if the intersection is larger than 50% of the union of the two region, it is considered a correspondence.

We note that this correspondence is called point-to-point correspondence as defined in [16]. It is different from region-to region correspondence as defined in [12] which considers only the second condition above. We take other widely used algorithms such as SIFT, SURF, ORB, BRISK and FREAK to make the comparison. All matching tests employ brute-force algorithm using floating distance for SIFT, SURF and Hamming distance for binary descriptors. For the sake of fairness, we set the same value for the number of relevant keypoints returned by detectors. This value is set to be 1000 keypoints in this test. As a reminder, the SIFT detector selects the relevant keypoints based on contrast thresholds and edge filter thresholds[1], whereas SURF uses Hessian response, ORB uses FAST score then Harris score. On the other hand, our algorithm HOOFR uses FAST score then Hessian score to refine keypoints for the detection.

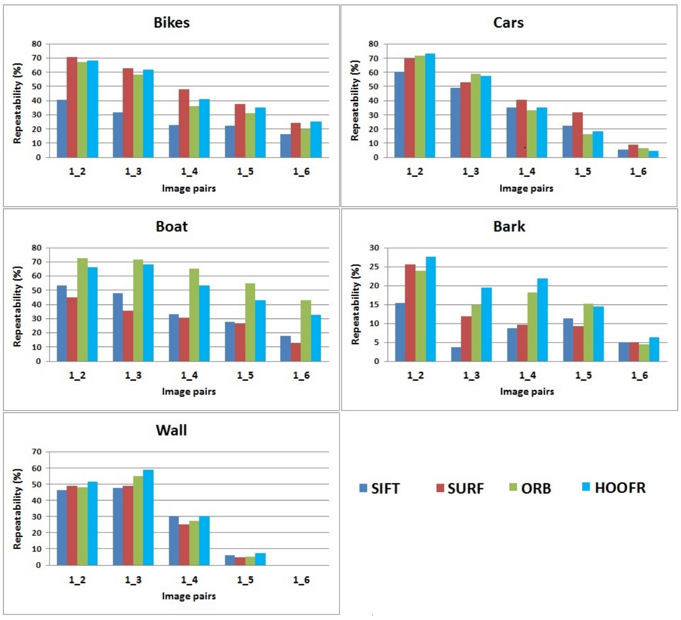


Fig. 7. Repeatability of detectors evaluated in image datasets.

A. HOOFR detector repeatability

The desirable property for a feature detector is repeatability. It represents the ability of a detector to find the same feature in two or more different images of the same scene. It is defined in [16] as the ratio between number of corresponding keypoints and the minimum number of points detected in the two images. We note that the number of points here is fixed to be 1000 for all detectors.

Figure 7 shows the repeatability evaluation on five transformations with independent characteristics. HOOFR exhibits a remarkable performance, and outperforms ORB on most of image sequences. This result underlines the conclusion of [12] that in general, Hessian matrix based detection outperforms detection based on the Harris matrix. The occasional low performance of SIFT is due in part to its sensitivity to rotation change and to blur (Boat, Bark and Bikes sequences); SURF exhibits competitive performance with respect to ORB and our algorithm HOOFR. Nevertheless, SURF is also time-consuming which limits its ability to be applied in real-time applications.

B. HOOFR binary descriptor comparison

Since we use the binary method to build the description, we compare HOOFR descriptor with other binary descriptor in the literature such as BRISK, ORB and FREAK. Recall vs 1-precision curve is used as proposed in FREAK [11] and BRISK [10] to judge the performances. Recall is defined as the ratio of number of correct matches/number of

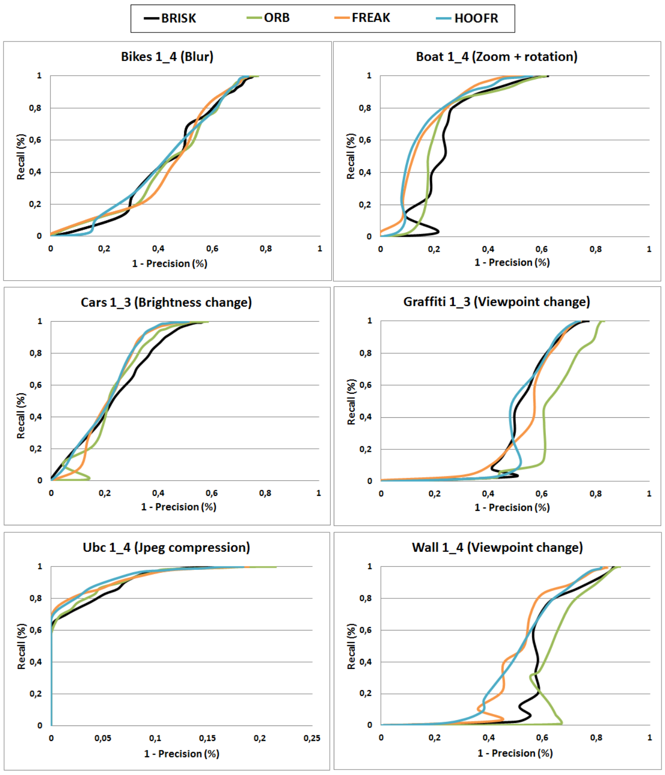


Fig. 8. Recall-precision for the evaluation of binary descriptors

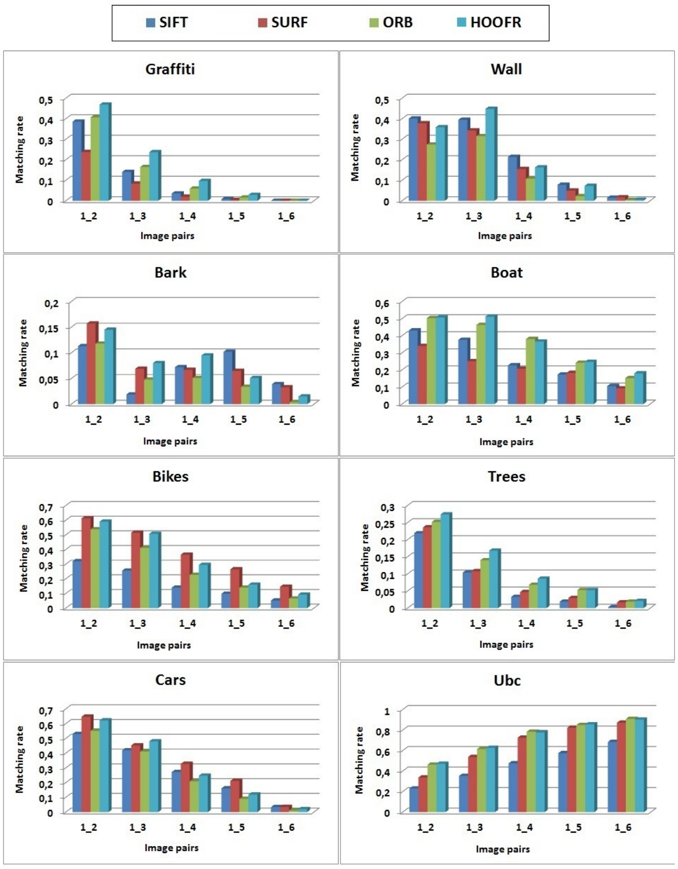


Fig. 9. Evaluation of matching rate in image datasets

correspondences, while 1-precision is the ratio of number of false matches/number of matches. In fact, the result of matching largely depends on the combination detector-descriptor. Nevertheless, the global ranking of matching performance of the descriptors remains the same regardless of the selected detector. Therefore, to ensure a fair comparison, we evaluate all descriptors by using the same detector. In this test, we chose ORB detector and the number of relevant keypoints returned is also 1000.

Figure 8 shows the recall-precision curves using thresholds based similarity matching of Hamming distance for a collection of images pairs from datasets. As the result of figure 8, HOOFR is generally more robust than FREAK. On the other hand, it overcomes ORB for all the tested image transformations. Moreover, despite the fluctuation in some cases, HOOFR has better performance than BRISK.

C. Overall evaluation of HOOFR

Our work proposes modifications in terms of detection and description at the same time, so we also evaluate the joint performance of both propositions compared to the well-known algorithms which have their own detector and descriptor such as SIFT, SURF or ORB. Due to the fact that SIFT and SURF use the floating descriptor while ORB and our work use binary descriptor, it does not make sense to use a similarity based method in matching. The reason is that similarity method highly depend on the threshold and it is difficult to determine equivalent value for the each type of descriptor. Therefore, in order to match two set of keypoints extracted from two images, for each keypoint in the first set, we simply select the keypoint in the second set which is the nearest neighbor (smallest matching distance). We present a factor called “Matching rate”(number of correspondences / number of matches) to compare the performances in this case.

In order to have a high matching score, an algorithm must exhibit a high detector repeatability and must concurrently have a high discrimination for the keypoint descriptor. As illustrated in figure 9, HOOFR performs competitively with SURF. It outperform SURF for the viewpoint change (Wall, Graffiti) or JPEG compression (Ubc), have a fluctuation for zoom-rotation (Bark, Boat) or blur (Bikes, Trees) and slightly fall behind SURF for brightness change(Cars). In contrast, HOOFR normally has overall better performance than SIFT and ORB.

D. Timings

Execution times have been recorded using a single core on a PC with Intel Core i7 3.4 GHz processor and 16GB RAM. Table I presents the results corresponding to detection of the first image in 4 selected sequences, while table II presents the description time for the same images. Moreover, table III shows the extraction time (detection+description) of the

TABLE I
DETECTION TIME (MILLISECONDS) OF DIFFERENT DETECTORS (1000 RELEVANT KEYPOINTS RETURNED)

	Bark_1	Graffiti_1	Boat_1	Wall_1
	(512x765)	(640x800)	(680x850)	(700x1000)
SIFT	860	919	1554	1722
SURF	129	137	169	202
ORB	34	44	79	107
HOOFR	33	42	76	105

TABLE II
DESCRIPTION TIME (MILLISECONDS) OF DIFFERENT DESCRIPTORS FOR 1000 KEYPOINTS

	Bark_1	Graffiti_1	Boat_1	Wall_1
	(512x765)	(640x800)	(680x850)	(700x1000)
SIFT	3611	3873	4024	4093
SURF	479	488	492	501
ORB	16	18	18	20
BRISK	23	24	24	24
FREAK	20	21	21	21
HOOFR	18	20	20	20

algorithms having its own detector and descriptor. The values are averaged over 50 runs.

Regarding the detector, the timings show an advantage of HOOFR. Its computation is even faster than ORB detector although the latter is the fastest detector currently available. The reason is that the Hessian response is time-saving to compute against Harris response.

In terms of description, we also clearly highlight the advantage of binary descriptors, with an order of magnitude faster than SURF and two order of magnitude faster than SIFT. Among the binary descriptors, FREAK is inspired by BRISK and it is more efficient than BRISK. Following the optimization trend, HOOFR is inspired by FREAK, and it is more robust, memory-saving and slightly faster than the original. We note that although the descriptor size and the number of pairs for orientation estimation were reduced in HOOFR in comparison to FREAK, we can not gain a significant acceleration due to more receptive fields being sampled (49 points) than in the case of FREAK (43 points). Hence, for each keypoint description, HOOFR takes more time to compute the Gaussian filter for all receptive fields. However, even though ORB is the fastest descriptor, in general, the extraction time (detection + description) of ORB is similar to that of HOOFR while our proposal maintains the better matching results.

TABLE III
EXTRACTION TIME (MILLISECONDS) OF DIFFERENT ALGORITHMS (DETECTION + DESCRIPTION) FOR1000 RELEVANT KEYPOINTS RETURNED

	Bark_1	Gratifi_1	Boat_1	Wall_1
	(512x765)	(640x800)	(680x850)	(700x1000)
SIFT	4471	4792	5578	5815
SURF	608	625	661	703
ORB	50	62	97	127
HOOFR	51	62	96	125

IV. CONCLUSION

We have presented a method named HOOFR, which aims to address the fundamental computer vision problem of detecting, describing and matching image keypoints. Our detector is the combination of ORB with a Hessian score, while our descriptor employs a human retina based descriptor consisting of a FREAK version with enhanced overlapping. Our proposal offers a better compromise between speed and matching quality against other state of the art algorithms. The experimental test shows that HOOFR exhibits competitive performance but much faster speed than SURF, SIFT. Besides, HOOFR exhibits comparably low computation cost as ORB, which it outperforms performance wise. In future work, we want to further investigate the applicability of HOOFR in different contexts, and to implement it efficiently on embedded heterogeneous systems (ARM-FPGA) which are adapted to real-time mobile applications.

REFERENCES

[1] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[2] S. Se, D. Lowe, and J. Little, “Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks,” *The international Journal of robotics Research*, vol. 21, no. 8, pp. 735–758, 2002.

[3] M. Brown and D. G. Lowe, “Recognising panoramas.” in *ICCV*, vol. 3, 2003, p. 1218.

[4] H. Bay, T. Tuytelaars, and L. Van Gool, “Surf: Speeded up robust features,” in *Computer vision–ECCV 2006*, 2006, pp. 404–417.

[5] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “Orb: an efficient alternative to sift or surf,” in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2564–2571.

[6] E. Rosten and T. Drummond, “Machine learning for high-speed corner detection,” in *Computer Vision–ECCV 2006*, 2006, pp. 430–443.

[7] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua, “Brief: Computing a local binary descriptor very fast,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 7, pp. 1281–1298, 2012.

[8] Y.-D. Kim, J.-T. Park, I.-Y. Moon, and C.-H. Oh, “Performance analysis of orb image matching based on android,” *International Journal of Software Engineering and Its Applications*, vol. 8, no. 4, pp. 11–20, 2014.

[9] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, “ORB-SLAM: a versatile and accurate monocular SLAM system,” *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[10] S. Leutenegger, M. Chli, and R. Y. Siegwart, “Brisk: Binary robust invariant scalable keypoints,” in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2548–2555.

[11] A. Alahi, R. Ortiz, and P. Vandergheynst, “Freak: Fast retina keypoint,” in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. Ieee, 2012, pp. 510–517.

[12] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool, “A comparison of affine region detectors,” *International journal of computer vision*, vol. 65, no. 1-2, pp. 43–72, 2005.

[13] E. Tola, V. Lepetit, and P. Fua, “Daisy: An efficient dense descriptor applied to wide-baseline stereo,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 5, pp. 815–830, 2010.

[14] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 886–893.

[15] F. Bianconi and A. Fernández, “Evaluation of the effects of gabor filter parameters on texture classification,” *Pattern Recognition*, vol. 40, no. 12, pp. 3325–3335, 2007.

[16] K. Mikolajczyk and C. Schmid, “Scale & affine invariant interest point detectors,” *International journal of computer vision*, vol. 60, no. 1, pp. 63–86, 2004.