

An optimisation approach to the design of a fast, compact and distinctive binary descriptor

Mariusz Oszust¹

Received: 23 March 2015 / Revised: 28 April 2016 / Accepted: 2 May 2016 / Published online: 12 May 2016
© The Author(s) 2016. This article is published with open access at Springerlink.com

Abstract The desired local feature descriptor should be distinctive, compact and fast to compute and match. Therefore, many computer vision applications use binary keypoint descriptors instead of floating-point, rich techniques. In this paper, an optimisation approach to the design of a binary descriptor is proposed, in which the detected keypoint is described using several, scale-dependent patches. Each such patch is divided into disjoint blocks of pixels, and then, binary tests between blocks' intensities, as well as their gradients, are used to obtain the binary string. Since the number of image patches and their relative sizes influence the descriptor creation pipeline, a simulated annealing algorithm is used to determine them, optimising recall and precision of keypoint matching. The simulated annealing is also used for dimensionality reduction in long binary strings. The proposed approach is extensively evaluated and compared with SIFT, SURF and BRIEF on public benchmarks. Obtained results show that the binary descriptor created using the resulted pipeline is faster to compute and yields comparable or better performance than the state-of-the-art descriptors under different image transformations.

Keywords Binary descriptor · Keypoints · Keypoint matching · Optimisation · Simulated annealing

1 Introduction

Recently, many computer vision applications for object recognition, tracking or scene reconstruction, have been using local feature descriptors. Such descriptors capture an image content that surrounds the detected keypoint and express it in the form of a vector. Feature descriptors with real-valued vectors produce the highest quality results, both in terms of robustness and performance, against a variety of image transformations. Scale-Invariant Feature Transform (SIFT) [1, 2] and Speeded Up Robust Features (SURF) [3] approaches are such floating-point descriptors. However, their computation time is long. Also, high dimensionality of compared vectors has negative impact on the image matching time. Therefore, binary descriptors have become more attractive in recent years, since they are compact and faster to compare using Hamming metric. In most cases, hand-crafted binary descriptors are obtained using pairwise tests between intensities of predefined parts of described image patch, i.e., pixels or regions according to a sampling pattern [4–8]. However, binary descriptors can be long, what requires an additional procedure for their reduction. The bit selection procedures use machine learning approaches [8–10] or optimisation [11, 12].

In this paper, an approach to the design of the binary descriptor is proposed. In the approach, 3.5 M SURF keypoints were detected on 4000 transformed images. Then, a simulated annealing (SA) algorithm [13, 14] determined a solution that maximised recall, as well as precision of keypoint matching. The solution consists of a set of image patches and their sizes. The patches are divided into disjoint pixel blocks. Finally, the binary string was created as a result of pairwise tests on intensities and gradients of blocks. Such binary string can be long; therefore, the SA was run in order to determine the most important 128 or 256 bits.

✉ Mariusz Oszust
marosz@kia.prz.edu.pl

¹ Department of Computer and Control Engineering, Rzeszow University of Technology, W. Pola 2, 35-959 Rzeszow, Poland

The main novelty of the proposed approach is the use of the SA for optimisation of parameters of the binary descriptor creation pipeline. The dimensionality reduction approach of the binary descriptor with this optimisation technique is also novel. The details of the descriptor and experimental results showing its promising performance are presented in the paper. SURF, SIFT and BRIEF [5] descriptors are evaluated for comparison.

The remaining part of the paper is organised as follows. Section 2 provides a presentation of the state-of-the-art approaches. A detailed description of the proposed method can be found in Sect. 3. Extensive evaluation of the descriptor on widely used image datasets is reported in Sect. 4. Section 5 concludes the paper.

2 Related work

A growing number of computer vision applications, incorporating different approaches to image content description, reveal a tendency to construct faster, and still maintaining desirable properties, solutions. Many of them are focused on interest point (keypoint) description [15–18]. Here, a keypoint is a meaningful pixel group which is repetitively detected in spite of different image transformations (e.g., illumination, viewpoint change, or rotation). SIFT [2] is the keypoint detector and the descriptor. This technique produces high-quality results, both in terms of robustness and performance against common image transformations. In SIFT, keypoints are found using difference of Gaussians applied in scale space. However, the keypoint detection approach present in SIFT requires time-consuming rescaling and smoothing of the image. Furthermore, high dimensionality of the descriptor (128 values) slows keypoint matching. Another, high-dimensional, real-valued descriptor that in some cases outperforms SIFT is SURF [3]. The method is based on SIFT, but it incorporates many simplifications reducing its complexity. In SURF, the orientation of the keypoint is extracted from a circular region with scale-dependant radius. The descriptor is created by union of vectors resulted from sums of horizontal and vertical Haar wavelet responses and their absolute values calculated for a square region centred at the keypoint. Here, the descriptor consists of 64 floating-point values.

In comparison with newly developed approaches, the performance of SIFT (and SURF) has not been significantly outperformed [18]. However, in order to speed up the keypoint description and to reduce the dimensionality of the descriptor, binary descriptors have been introduced. Most of them apply pairwise tests between intensities of image regions or pixels according to some sampling pattern. Intensity comparison makes these solutions robust to photometric changes. Binary Robust Invariant Scalable Keypoints

(BRISK) [4] uses AGAST [19] corner detector to find keypoints; then, a circular pattern with equally spaced points is used in binary tests. In this technique, the orientation of the keypoint is estimated using local gradients between the sampling pairs. In Binary Robust Independent Elementary Features (BRIEF) [5], the binary vector is obtained in tests between the intensities of point pairs along the same lines. Points are selected randomly from an isotropic Gaussian distribution. Since it is not rotation invariant, Oriented FAST and Rotated BRIEF (ORB) [6] was developed. The sampling pairs in ORB are determined in a learning process involving a greedy algorithm working on 300 K keypoints, maximising the amount of information carried by the descriptor. Here, FAST [20] detects keypoints. The dominant orientation in ORB is obtained using the intensity centroid approach [21]. In spite of introduced improvements, ORB performs better than BRIEF only in presence of large rotation or scale change [17]. In Fast Retina Keypoint (FREAK) [7], the sampling pattern was inspired by the human visual system. Here, the learning process is similar to ORB's with additional rejection of correlated pairs. The orientation of the keypoint is estimated using local gradients of 45 sampling pairs. FREAK seems to be outperformed by other binary descriptors according to [22, 23]. In recently developed Local Difference Binary (LDB) [8, 11] descriptor, in turn, the image patch of a predefined size (e.g., 45×45 , 50×50 pixels [11]) is divided into 4, 9, 16 and 25 square cells. Then, binary tests between values representing cells are applied. In this binary descriptor, gradients are compared, as well as cells' mean intensities. AdaBoost-based important bit selection method, working on 400 K patches from [24], is applied to produce 256 bit binary strings. The orientation of the keypoint is estimated using the intensity moments. In works [9, 10], a supervised learning framework that finds low-dimensional descriptors is proposed. Here, the content of the image patch is modelled using local nonlinear filters that are selected with boosting. Boosting also helps in selection of the most important bits. In another approach [12], the descriptor design and its dimensionality reduction are formulated as convex optimisation problems involving separation of positive and negative patches. The solutions are found using support vector machine (SVM) solvers.

3 The approach

In the following subsections, the proposed binary descriptor is presented. At first, its design is formulated as an optimisation problem. Then, a simulated annealing algorithm is used to determine its creation pipeline. Finally, the simulated annealing is used for dimensionality reduction in long binary strings.

3.1 Optimisation problem

The proposed approach is partially inspired by Multiscale Block Local Binary Pattern (MBLBP) [25], in which authors propose to compare average intensities of neighbouring blocks of pixels in order to create a binary string. In MBLBP, three patches with 3×3 , 9×9 , and 15×15 pixels are divided into nine blocks (cells). MBLBP is a dense descriptor typically applied to all pixels in the image. Here, the binary string is created by concatenation of strings obtained for three patches. The pairwise tests are performed only between some cells. It is worth mentioning that BRISK descriptor was also inspired by the dense approach (DAISY [26]). MBLBP uses only three image patches, and their sizes are given in pixels. The similar limitation can be seen in [11]. The approach proposed in this paper follows the idea of nine cells (pixel blocks) for the image patch. Moreover, the number of image patches and their sizes for the pipeline of creation of the descriptor are found in optimisation experiments. The proposed solution also uses intensity gradients inspired by SURF.

The variability in the pipeline of the creation of the descriptor caused by the number of used patches and their size expressed in terms of relation between the length of patch's side and the keypoint's scale, has led to the formulation of the following optimisation problem. In the problem, precision and recall of keypoint matching were maximised. At the beginning, a keypoint detector generates M_i keypoints for an input image i , $i = 1, \dots, N$, where N denotes the number of input images. The keypoint K_i^k ($K_i^k = 1, \dots, K_i^{M_i}$) is described with the help of P image patches. Each patch p ($p = 1, \dots, P$) has its keypoint's scale multiplier S_p and nine square pixel blocks B_p^j , $j = 1, \dots, 9$. The S_p determines the size of the patch, i.e., the patch's side in pixels is equal to S_p multiplied by the keypoint's scale. The B_p^5 -th block is centred at the keypoint's location (x, y) . In order to create the binary descriptor, pairwise tests are performed between blocks' intensities. Such test returns *true* (1) if a difference of values is smaller than 0, and *false* (0), otherwise. Since blocks for the p -th patch have the same number of pixels, sums of their intensities ($I(B_p^j)$) can be compared. This solution is motivated by efficient calculation of such sums using the integral image technique [3,27]. Also, the first-order intensity gradients in the x and y directions ($D_x(B_p^j)$ and $D_y(B_p^j)$) are computed using Haar-like box filters and the integral image, as in SURF [3]. Gradients for B_p^j -th block are obtained for its central pixel. The results of pairwise tests performed on gradients, in a form of the binary vector, are added to binary strings obtained for intensity I . Finally, for the p -th patch 108 binary tests are performed, and for the K_i^k -th keypoint 108 P -bit string is obtained. Exemplary image patches and their division are presented on Fig. 1.

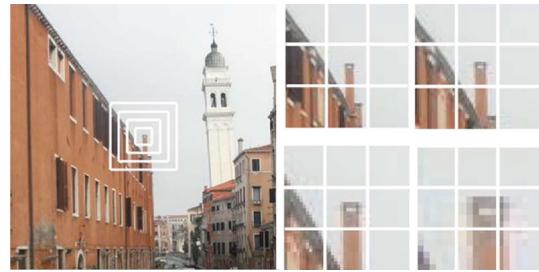


Fig. 1 An exemplary four image patches centred at the keypoint and their division

The number of image patches P and multipliers S can be seen as decision variables in the optimisation problem. Since *precision* and *recall* measures [17] reflect the quality of performance of a given local feature descriptor in image matching tests, they can be used as an objective function. Precision counts correctly matched pairs out of all returned matches, while recall, in turn, counts such pairs out of corresponding pairs. Higher values of both metrics are considered better; therefore, their multiplication was used as the objective function in the proposed approach.

3.2 Descriptor design using simulated annealing

The simulated annealing algorithm (SA) [13, 14] reflects phenomena observed while slow cooling of molten metals. The annealing process achieves the global minimum state using random fluctuations in energy. Such fluctuations help the process to escape local minima. The SA algorithm requires definitions of: (1) a vector of decision variables x , (2) an objective function $F_C(x)$, (3) a method $G(x)$ for generation of a neighbouring solution x' , based on information stored in x , (4) a temperature drop T , and (5) an acceptance criteria of a weaker solution $F_A(x, x', T)$. The temperature drop T can be seen as the number of iterations of the SA. The algorithm consists of the following steps:

1. Create random x
2. Calculate $F_C(x)$
3. While $T > T_{stop}$ then:
 - (a) Create $x' = G(x)$
 - (b) Calculate $F_C(x')$
 - (c) If $F_C(x')$ is better than $F_C(x)$ or $F_A(x, x', T)$ accept $x', x = x'$
 - (d) Decrease T .

F_A is calculated as follows:

$$F_A(x, x', T) = e^{-\frac{\Delta F_C}{T}}, \quad (1)$$

where $\frac{\Delta F_C}{T}$ expresses the difference between values of $F_C(x)$ and $F_C(x')$. T is iteratively decreased, at the beginning causing that weaker solutions to have a higher chance to be accepted, i.e., if F_A is larger than a randomly selected number in the range $[0, 1)$, the solution x' is accepted.

The problem of determination of the descriptor creation pipeline is solved using the SA, x in this case refers to finding P and S . For the important bit selection problem, or dimensionality reduction, the solution x takes a form of a binary vector indicating which bits are used while maximising $F_C(x)$ and reducing the descriptor's binary string to a given length (e.g., 128, 256 bits). F_C is given as

$$F_C(x) = \sum_{i=1}^N P_i R_i, \quad (2)$$

where P denotes precision, R recall, and i is the input image. The $G(x)$ function changes values in S_p and adds or rejects the patch p . In the dimensionality reduction problem, $G(x)$ selects a predefined number of the most important bits randomly changing information carried by x .

The choice of the SA was motivated by its simplicity, in terms of implementation, and the small number of parameters. Furthermore, the proposed objective function does not require a process of preparation of negative examples, as in approaches with SVM or boosting. Population-based optimisation algorithms (e.g., genetic, immune, or swarm) were also taken into consideration. They evaluate and reject many solutions in iterations, since each solution requires time (and memory)-consuming computation of the objective function; the SA seems to be a reasonable approach, providing acceptable results with fewer objective function calls.

Once P and S are obtained, K_i^k keypoints are generated by a keypoint detector on the input image i . It is assumed that the detector provides keypoints with scales. Then, the keypoints are described in the following manner:

1. For each keypoint K_i^k
 - (a) Select $p = 1, \dots, P$ patches with size $S_p\sigma \times S_p\sigma$, where σ denotes keypoint's scale (in pixels).
 - (b) For each patch
 - i. For each B_p^j block, $j = 1, \dots, 9$, within the p -th patch: Calculate $I(B_p^j)$, $D_x(B_p^j)$, and $D_y(B_p^j)$ associated with the centre of the block.
 - ii. Perform $9!/(2! \times 9!) = 36$ pairwise tests for $I(B_p^j)$, $D_x(B_p^j)$ and $D_y(B_p^j)$.
 - (c) Concatenate $108P$ binary strings into the descriptor of the keypoint K_i^k .

The dominant orientation of the detected interest point is estimated using wavelet responses in horizontal and vertical direction. A similar step is present in SURF, but here

only half-wavelet responses are used. The orientation helps to determine the position of the central pixel of each outer block (B_p^j , $j \neq 5$).

4 Experiments

In the following subsections, experiments on the design of the pipeline for the creation of the descriptor and the most important bit selection are presented. Then, the resulting descriptor family is evaluated on public benchmarks.

4.1 Optimisation

For the design of the descriptor's pipeline, 1K images were randomly selected from 25K *MIRFLICKR* dataset [28]. The dataset consists of images downloaded from the social photography site *Flickr*. They were rotated (90° and 45°) and scaled ($1/2$ and $2/3$). Then, 225 images from *Phos* dataset [29] were added to the resulting dataset. *Phos* contains 15 scenes captured under different illumination conditions (i.e., uniform and nonuniform). Each scene has selected one base image. It can be seen that the number of images being under different illumination changes is significantly smaller than rotated or scaled images. Since constructed binary descriptor uses pairwise intensity tests, it is, like other binary descriptors, robust against illumination changes. Therefore, this part of the learning dataset was smaller.

3,519,420 keypoints were detected by SURF on 5225 images, prior to the optimisation experiments. Integral images for the learning dataset, as well as lists of keypoints have large memory requirements, thus only one solution, x , was evaluated at a time, i.e., in terms of calculation of the objective function F_C on used CPU with 16 GB RAM. The SA maximises F_C (see Eq. (2)) which uses precision and recall metrics. For a given input image, precision was calculated as the number of correct matches, i.e., the number of corresponding pairs of keypoints between the base image and the distorted one, to the number of all returned pairs. The match was accepted if a distance ratio between the first and the second close described keypoint on the second image was below or equal a threshold of 0.8 [2, 17] and the location of the keypoint on the second image was within three pixels from the expected location. The expected location of the keypoint was inferred using homography [17]. 4K distorted images from *MIRFLICKR* were matched with the original 1K images, and 15 reference images were selected from *Phos* for the same purpose. Since the dataset consists of 4210 pairs of images and used metrics have maximum value equal 1, the best (maximal) possible objective function value is 4210. This would indicate the case in which each detected keypoint was successfully matched. Due to image transformations, such large F_C value was not likely to be found.

Table 1 Objective function values obtained in experiments

Optimisation problem	Maximum	Minimum	Mean	SD
Descriptor design	2119.1	2093.3	2106.3	5.6
Selection of 256 bits	2069.9	2060.3	2064.3	2.9
Selection of 128 bits	2004.1	1975.1	1987.5	7.4

The SA was run 30 times with $T_{start} = 1$. The T was decreased using $T_{next} = 0.9T_{old}$ in 100 iterations of the algorithm. The maximal possible number of patches P was set to 15, values of scale multipliers S_p were selected in range 1 to 150 with step 1. Scale multipliers could be floating-point values but the use of integers significantly reduced the search space of the SA algorithm. Table 1 contains mean, maximal, minimal and standard deviation of the objective function of solutions obtained in experiments. The best binary descriptor was the input to the second optimisation problem of finding the most 128 or 256 important bits. The dimensionality of the binary vector was reduced since it affects the matching time and the storage cost. Parameters of the SA for the second problem remained unchanged. It can be seen, the obtained mean values are promising, and the results are characterised by small standard deviations. The best result ($F_C(x_{best}) = 2119.1$) was obtained for the set of $P = 12$ patches with the following multipliers $S_p = [7, 10, 16, 19, 23, 28, 30, 34, 44, 56, 69, 89]$. This gives the descriptor with 1296 bits. For comparison, objective function calculated for SURF approach was equal to 1642.2; BRIEF obtained $F_C = 136.7$ and SIFT 2889.6. SIFT used its own detector to generate keypoints, thus about half of the number of SURF keypoints were detected. The proposed descriptor using SIFT keypoints yielded $F_C = 2744.3$. They seem to be more suitable, but in experiments previously detected SURF points remained, since their number was larger, making F_C optimisation more difficult.

The creation of the descriptor using twelve patches can be time-consuming. Therefore, a simplified version of the descriptor was used. It contains only four patches (432 bits) with $S_p = [5, 10, 15, 20]$, and obtained $F_C = 1683.9$ with SURF keypoints.

4.2 Evaluation

Evaluation of the proposed descriptor was made using two publicly available, widely used, Mikolajczyk et al. [15] (*Oxford*) and Heinly et al. [17] datasets. They contain image sequences with known homography between the first image in the sequence of images and the others. There are six to nine images in each sequence. Images exhibit an increasing amount of transformations, such as rotation (sequences: *Ceiling* [17], *Rome* [17], *Semper* [17]), scaling (*Venice* [17]),

rotation with scaling (*Bark* [15], *Boat* [15]), view point change (*Wall* [15], *Graffiti* [15]), blur (*Bikes* [15], *Trees* [15]), illumination (*Day and night* [17]), exposure (*Leuven* [15]) and JPEG compression (*Ubc* [15]).

SURF, SIFT and BRIEF were selected for comparison due to their high performance reported in many works. BRIEF (512 bit) is representative of binary descriptors, and its performance was confirmed in work [17], despite being not fully rotation and scale invariant. The proposed descriptor, since then called Optimised Binary Robust fAst Features (OBRAF), is represented by its three versions, with 1296, 256, 128-bit vectors (twelve patches) and simple 432-bit version with four patches (nonoptimised BRAF). For each image pair, about 1 K SURF keypoints were detected and described. SIFT generated similar number of keypoints and described them, since, in the spirit of fairness, SIFT descriptor should be coupled with its own detector. Keypoints were matched using the same protocol as in the experiments with the optimisation. For comparative analysis, precision and putative match ratio [17] were used. Precision of matching has a practical significance, since all returned pairs often undergo further processing, e.g., using RANSAC [30]. Putative match ratio (PMR) counts how many detected keypoints were returned as matched. This measure is influenced by the descriptor's ability to differentiate similar image regions, and could be understood as a measure of distinctiveness of the descriptor. In some cases, the keypoints can be detected close to each other.

Figure 2 shows mean values of the precision and PMR results obtained for both datasets. For all 13 image sequences, OBRAF family of descriptors was better, in terms of precision values, than one of the compared floating-point descriptors. For the PMR metric, OBRAF descriptors were better than one of the floating-point counterparts for seven image sequences. BRIEF was outperformed by the other methods, even in tests without rotation and scale changes. The simplified version of the proposed descriptor, BRAF, in some cases performed better than SIFT or SURF. Since mean values were presented, the behaviour of the descriptors cannot be seen in the most difficult tests, i.e., when the metrics are evaluated for the last images in the sequences. Therefore, *Recall versus 1-Precision* curves obtained for the last images in sequences, considered as the most challenging, are shown on Fig. 3. Here, the proposed descriptor family achieved comparative or better performance than floating-point descriptors. In the *Night and day* sequence, BRIEF performed quite well, since it was designed for such transformation.

Table 2 contains comparison of descriptor computation time. Since most descriptors in original works were computed on similar or faster machines using different numbers of keypoints, the table contains timings per keypoint. OBRAF is implemented in Java as a single-threaded appli-

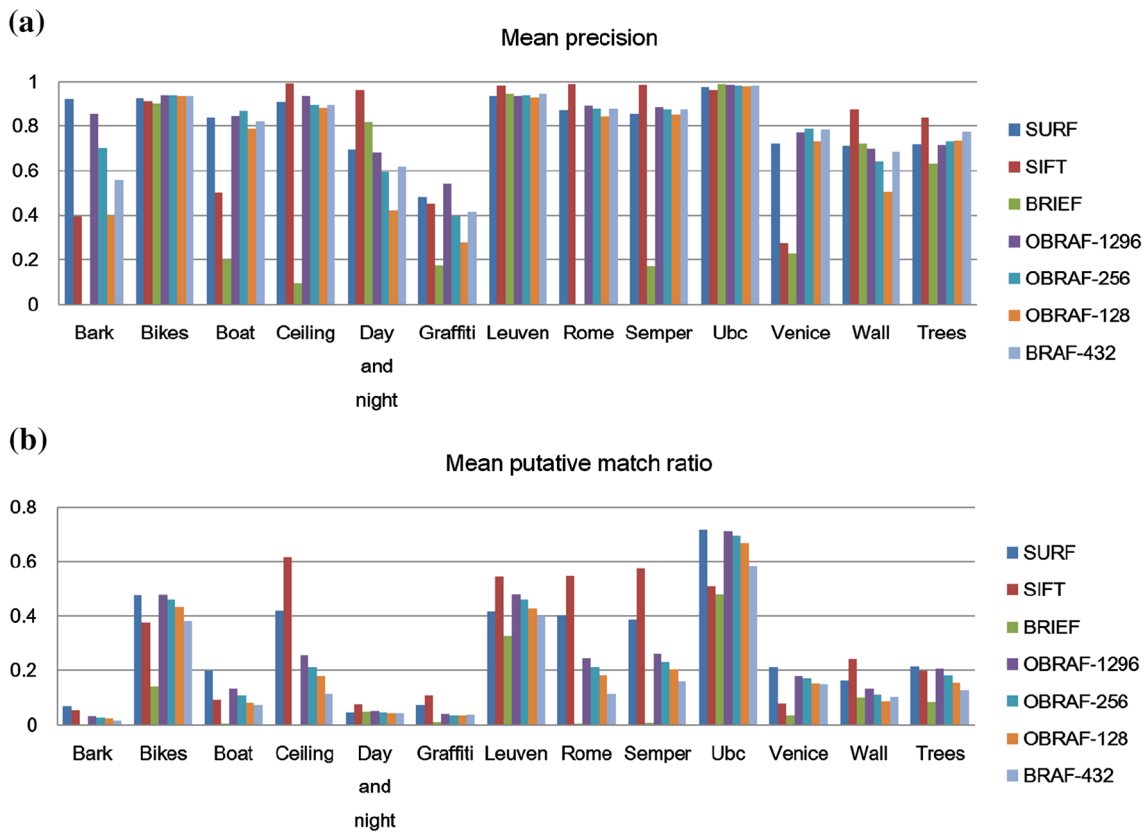


Fig. 2 Mean precision (a) and mean putative match ratio (b) for descriptors evaluated on Mikolajczyk et al. [15] and Heinly et al. [17] datasets

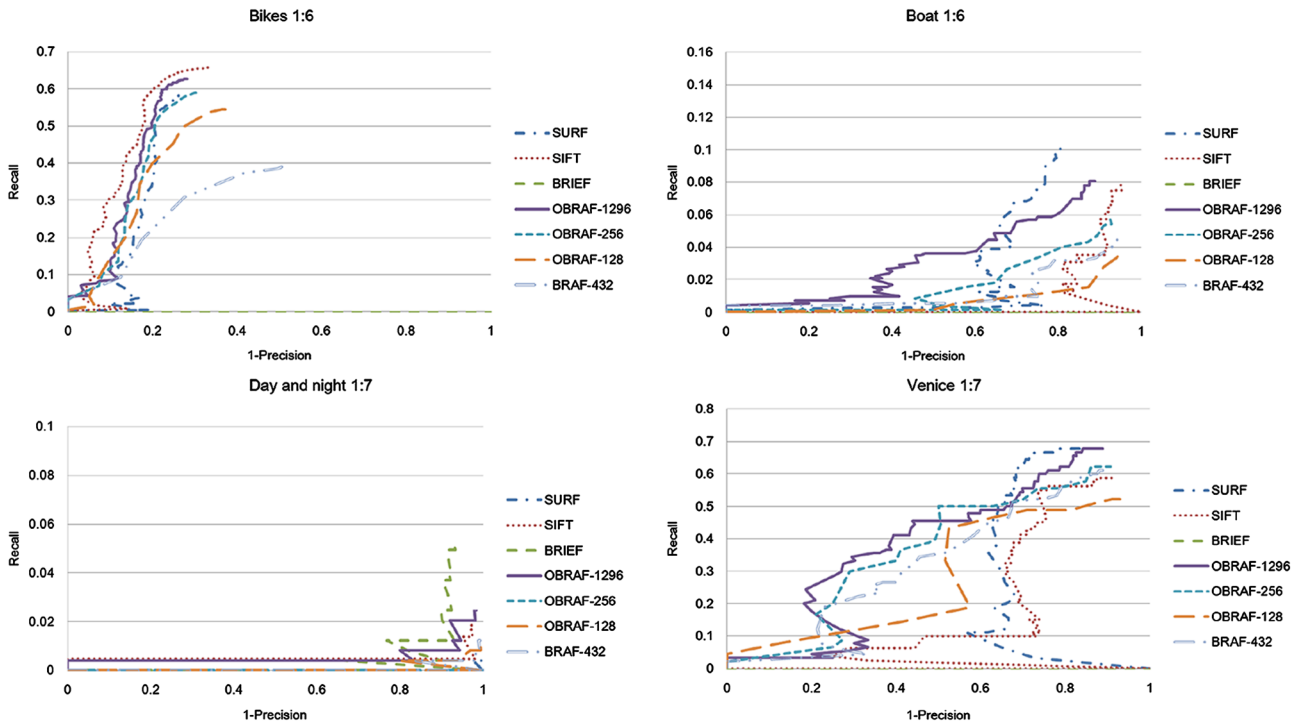


Fig. 3 Recall versus 1-Precision curves for compared descriptors evaluated on Mikolajczyk et al. [15] and Heinly et al. [17] datasets. Curves were obtained using the last image in a given image sequence. The name

of the sequence is given above the curve, the pair of images is denoted as 1 : t , where t is the number of the last image, and 1 indicates the number of the reference image

Table 2 Descriptor computation time (per keypoint)

Name	Length (bits)	Computation time (ms)
OBRAF	1296, 256, 128	0.027* [⊖] , 0.044*
BRAF	432	0.011* [⊖] , 0.026*
SIFT	1024	6.156 [4], 0.407*
SURF	512	0.43 [4], 0.66 [5], 0.67 [8], 0.15*
BRISK	512	0.037 [4], 0.026 [8]
BRIEF	512	0.0187 [⊖] [5], 0.053* [⊖]
ORB	256	0.056 [8]
LDB	256	0.048 [8]
FREAK	512	0.039 [8]

Results with * were obtained in experiments reported in this paper (Java implementations), [⊖] denotes an approach, in which the orientation of the keypoint is not estimated

cation, and all presented experiments were run on Intel Core i7-2720QM 2.2GHz, 16GB of RAM, Microsoft Windows 7, and Java 8.0. Therefore, BoofCV [31] (Java) implementations of SIFT, SURF and BRIEF were used. SURF in BoofCV library is faster and produce better results than many widely used SURF implementations, e.g., *OpenSurf*, *ETH* or *OpenCV* [31]. It can be seen that OBRAF descriptor family obtained similar or better computation time than other descriptors. A simplified OBRAF version (BRAF), in which the dominant orientation is not estimated, is computed in 0.011 ms, 0.5 times faster than BRIEF that also does not incorporate this operation. It significantly outperforms other compared approaches. Cited descriptors were implemented in C++, and their computation time was mostly optimised. OBRAF's computation consists of many steps which are independent, e.g., each patch and each block within the patch can be processed independently what makes the descriptor easy to parallel. Therefore, some further improvements in shortening the computation time are expected.

The matching time is strongly affected by the descriptor length, and binary vectors can be efficiently compared (Hamming distance) on modern CPUs using binary *XOR* and *population count* instructions. The state-of-the-art sparse binary descriptors are seldom shorter than 128 bits. Since 128 and 256-bit OBRAF versions yielded promising performance results, they would also offer the state-of-the-art matching time.

5 Conclusions

In this paper, a novel descriptor, OBRAF, is proposed as a result of an optimisation approach to the design of the descriptor creation pipeline. The SA algorithm was used to solve two problems. At first, it determined the number of patches and their sizes used for the description of the inter-

est point. Then, the solution of the dimensionality reduction problem was found. In both cases, recall and precision of keypoint matching were used as the objective function. The obtained descriptor family was evaluated on two popular image datasets. Experimental results showed that OBRAF is faster than the state-of-the-art descriptors while maintaining comparable or better performance under different image transformations.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Lowe, D.G.: Object recognition from local scale-invariant features. In: Proceedings of the 7th International Conference Computer Vision, vol. 2, pp. 1150–1157. Kerkyra, Sept 20–27 (1999). doi:[10.1109/ICCV.1999.790410](https://doi.org/10.1109/ICCV.1999.790410)
- Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision* **60**(2), 91–110 (2004). doi:[10.1023/B:VISI.0000029664.99615.94](https://doi.org/10.1023/B:VISI.0000029664.99615.94)
- Bay, H., Ess, A., Tuytelaars, T., Gool, L.V.: SURF: speeded-up robust features. In: Proceedings of the European Conference Computer Vision, pp. 404–417. Graz, May 7–13 (2006). doi:[10.1007/11744023_32](https://doi.org/10.1007/11744023_32)
- Leutengger, S., Chli, M., Siegwart, R.Y.: BRISK: binary robust invariant scalable keypoints. In: Proceedings of the International Conference on Computer Vision, pp. 2548–2555. Barcelona, Nov 6–13 (2011). doi:[10.1109/ICCV.2011.6126542](https://doi.org/10.1109/ICCV.2011.6126542)
- Calonder, M., Lepetit, V., Strecha, C., Fua, P.: BRIEF: binary robust independent elementary features. In: Proceedings of the European Conference on Computer Vision, pp. 778–792. Heraklion, Sept 5–11 (2010). doi:[10.1007/978-3-642-15561-1_56](https://doi.org/10.1007/978-3-642-15561-1_56)
- Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: ORB: an efficient alternative to SIFT or SURF. In: Proceedings of the International Conference on Computer Vision, pp. 2564–2571. Barcelona, Nov 6–13 (2011). doi:[10.1109/ICCV.2011.6126544](https://doi.org/10.1109/ICCV.2011.6126544)
- Alahi, A., Ortiz, R., Vandergheynst, P.: Freak: fast retina keypoint. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition, pp. 510–517. Providence, June 16–21 (2012). doi:[10.1109/CVPR.2012.6247715](https://doi.org/10.1109/CVPR.2012.6247715)
- Yang, X., Cheng, K.T.: Local difference binary for ultra-fast and distinctive feature description. *IEEE Trans. Pattern Anal.* **36**(1), 188–194 (2014). doi:[10.1109/TPAMI.2013.150](https://doi.org/10.1109/TPAMI.2013.150)
- Trzcinski, T., Christoudias, M., Lepetit, V., Fua, P.: Learning Image Descriptors with the Boosting-Trick. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) *Advances in Neural Information Processing Systems 25*. Proceedings of Neural Information Processing Systems Conference (NIPS), pp. 269–277. Curran Associates, Inc. (2012)
- Trzcinski, T., Christoudias, M., Fua, P., Lepetit, V.: Boosting binary keypoint descriptors. In: Proceedings of the International Conference IEEE CVPR, pp. 2874–2881. Portland, June 23–28 (2013). doi:[10.1109/CVPR.2013.370](https://doi.org/10.1109/CVPR.2013.370)
- Yang, X., Cheng, K.T.: LDB: an ultra-fast feature for scalable augmented reality on mobile devices. In: International Symposium

- IEEE Mixed and Augmented Reality, pp. 49–57. Atlanta, Nov 5–8 (2012). doi:[10.1109/ISMAR.2012.6402537](https://doi.org/10.1109/ISMAR.2012.6402537)
12. Simonyan, K., Vedaldi, A., Zisserman, A.: Learning local feature descriptors using convex optimisation. *IEEE Trans. Pattern Anal.* **36**(8), 1573–1585 (2014). doi:[10.1109/TPAMI.2014.2301163](https://doi.org/10.1109/TPAMI.2014.2301163)
 13. Kirkpatrick, S., Vecchi, M.P.: Optimization by simulated annealing. *Science* **220**(4598), 671–680 (1983). doi:[10.1126/science.220.4598.671](https://doi.org/10.1126/science.220.4598.671)
 14. Goffe, W.L., Ferrier, G.D., Rogers, J.: Global optimization of statistical functions with simulated annealing. *J. Econom.* **60**(1), 65–99 (1994). doi:[10.1016/0304-4076\(94\)90038-8](https://doi.org/10.1016/0304-4076(94)90038-8)
 15. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Van Gool, L.: A comparison of affine region detectors. *Int. J. Comput. Vis.* **65**, 43–72 (2005). doi:[10.1007/s11263-005-3848-x](https://doi.org/10.1007/s11263-005-3848-x)
 16. Xie, X., Mirmehdi, M.: *A Galaxy of Texture Features*. Handbook of Texture Analysis. Imperial College Press, London (2008)
 17. Heinly, J., Dunn, E., Frahm, J.M.: Comparative evaluation of binary features. In: Proceedings of the European Conference Computer Vision, pp. 759–773. Florence, Oct 7–13 (2012). doi:[10.1007/978-3-642-33709-3_54](https://doi.org/10.1007/978-3-642-33709-3_54)
 18. Miksik, O., Mikolajczyk, K.: Evaluation of local detectors and descriptors for fast feature matching. In: Proceedings of the 21st International Conference on IEEE Pattern Recognition, pp. 2681–2684. Tsukuba, Nov 11–15 (2012)
 19. Mair, E., Hager, G.D., Burschka, D., Suppa, M., Hirzinger, G.: Adaptive and generic corner detection based on the accelerated segment test. In: Proceedings of the European Conference on Computer Vision, pp. 183–196. Crete, Sept 5–11 (2010). doi:[10.1007/978-3-642-15552-9_14](https://doi.org/10.1007/978-3-642-15552-9_14)
 20. Rosten, E., Porter, R., Drummond, T.: Faster and better: a machine learning approach to corner detection. *IEEE Trans. Pattern Anal.* **32**, 105–119 (2010). doi:[10.1109/TPAMI.2008.275](https://doi.org/10.1109/TPAMI.2008.275)
 21. Rosin, P.L.: Measuring corner properties. *Comput. Vis. Image Underst.* **73**(2), 291–307 (1999). doi:[10.1006/cviu.1998.0719](https://doi.org/10.1006/cviu.1998.0719)
 22. Canclini, A., Cesana, M., Redondi, A., Tagliasacchi, M., Ascenso, J., Cilla, R.: Evaluation of low-complexity visual feature detectors and descriptors. In: Proceedings of the 18th International Conference on IEEE Digital Signal Processing, pp. 1–7. Fira, July 1–3 (2013). doi:[10.1109/ICDSP.2013.6622757](https://doi.org/10.1109/ICDSP.2013.6622757)
 23. Caetano, C., Avila, S., Guimaraes, S., Araujo, A.D.A.: Representing local binary descriptors with bossaNova for visual recognition. In: Proceedings of the 29th Annual ACM Symposium on Applied Computing, pp. 49–54. Gyeongju, March 24–28 (2014). doi:[10.1145/2554850.2555058](https://doi.org/10.1145/2554850.2555058)
 24. Winder, S., Hua, G., Brown, M., Picking the best DAISY. In: Proceedings of the International Conference on IEEE CVPR, pp. 178–185. Miami, June 20–25 (2009). doi:[10.1109/CVPR.2009.5206839](https://doi.org/10.1109/CVPR.2009.5206839)
 25. Liao, S., Zhu, X., Lei, Z., Zhang, L., Li, S.Z.: *Learning Multi-scale Block Local Binary Patterns for Face Recognition LNCS*, vol. 4642, pp. 828–837. Springer, New York (2007). doi:[10.1007/978-3-540-74549-5_87](https://doi.org/10.1007/978-3-540-74549-5_87)
 26. Tola, E., Lepetit, V., Fua, P.: Daisy: an efficient dense descriptor applied to wide-baseline stereo. *IEEE Trans. Pattern Anal.* **32**(5), 815–830 (2010). doi:[10.1109/TPAMI.2009.77](https://doi.org/10.1109/TPAMI.2009.77)
 27. Crow, F.C.: Summed-area tables for texture mapping. In: Proceedings of the 11th Conference on Computer Graphics and Interactive Techniques, pp. 207–212. ACM, Minneapolis, July 23–27 (1984). doi:[10.1145/964965.808600](https://doi.org/10.1145/964965.808600)
 28. Huiskes, M.J., Lew, M.S.: The MIR flickr retrieval evaluation. In: Proceedings of the 1st International Conference on ACM Multimedia Information Retrieval, pp. 39–43. Vancouver, Oct 30–31 (2008). doi:[10.1145/1460096.1460104](https://doi.org/10.1145/1460096.1460104)
 29. Vonikakis, V., Chrysostomou, D., Kouskouridas, R., Gasteratos, A.: A biologically inspired scale-space for illumination invariant feature detection. *Meas. Sci. Technol.* **24**(7), 074024 (2013). doi:[10.1088/0957-0233/24/7/074024](https://doi.org/10.1088/0957-0233/24/7/074024)
 30. Fischler, A.M., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **24**(6), 381–395 (1981). doi:[10.1145/358669.358692](https://doi.org/10.1145/358669.358692)
 31. Abeles, P.: Speeding up SURF. In: Proceedings of the 9th International Symposium Advances in Visual Computing, LNCS, vol. 8034, pp. 454–464. Springer, Rethymnon, July 29–31 (2013). doi:[10.1007/978-3-642-41939-3_44](https://doi.org/10.1007/978-3-642-41939-3_44)