A Binary Representation for Real-Valued, Local Feature Descriptors

Submitted: 8th October 2016; accepted: 10th January 2017

Mariusz Oszust

DOI: 10.14313/JAMRIS_1-2017/1

Abstract:
The usage of real-valued, local descriptors in computer vision applications is often constrained by their large memory requirements and long matching time. Typical approaches to the reduction of their vectors map the descriptor space to the Hamming space in which the obtained binary strings can be efficiently stored and compared. In contrary to such techniques, the approach proposed in this paper does not require a data-driven binarisation process, but can be seen as an extension of the floating-point descriptor computation pipeline with a step that allows turning it into a binary descriptor. In this step, binary tests are performed on values determined for pixel blocks from the described image patch. In the paper, the proposed approach is described and applied to two popular real-valued descriptors, SIFT and SURF. The paper also contains a comparison of the approach with state-of-the-art binarisation techniques and popular binary descriptors. The results demonstrate that the proposed representation for real-valued descriptors outperforms other methods on four demanding benchmark image datasets.

Keywords: *SIFT, SURF, LDAHash, binary tests, image matching, image recognition*

1. Introduction

Real-valued, local feature descriptors, such as Scale-Invariant Feature Transform (SIFT) [24] and Speeded Up Robust Features (SURF) [5] have already found their place in many computer vision applications, e.g., recognition [14], localisation [11, 12], tracking [8, 34], simultaneous localisation and mapping [17], or retrieval [6,15]. However, there is a need for the development of more efficient techniques, in terms of computation time, storage requirements, or robustness [6, 10, 12, 13, 25, 27]. Since SIFT and SURF are among the best performing floating-point descriptors [21, 26], there are many works which aim to preserve their distinctive properties while performing a dimensionality reduction of their long vectors [19], or storing them as binary strings after mapping the descriptor space into the Hamming space [35]. A transformation to binary strings allows for faster feature matching, since such strings can be efficiently compared on modern CPUs. For example, in [35] LDAHash was introduced, in which a projection matrix is selected and computed minimising in-class covariance and maximising covariance across classes of SIFT features. Then, a threshold vector is used for binarisation of projections and providing binary strings which maximise recognition rate. In another data-driven work [31], a vector of medians was used for binarisation of SIFT keypoints. The same approach can be easily applied to SURF vectors. The recently introduced Bi-DCT descriptor for dense matching [22] converts floating-point vectors into binary strings, taking into account DCT coefficients. The method exploits frequency and orientation information based on 2D DCT. In [16], in turn, dimensionality reduction of SIFT descriptor was performed using matched descriptor pairs and a linear discriminant embedding. PCA-SIFT technique [19] produces shorter real-valued descriptors based on principal component analysis (PCA) which is applied to SIFT keypoints. A more recent technique, shown in [18], uses a spectral embedding for transformation of the original feature space into an low-dimensional space, preserving the manifold structure and a relevance relationship among the images.

The approach introduced in this paper transforms high-dimensional information on the described image patch centred at the keypoint to a binary string. It is devoted to hand-crafted floating-point techniques for which the patch division into smaller pixel regions, or blocks, is easy accessible. In a typical binary descriptor, binary tests between intensities of pairs of points within the image patch are performed. This can be seen in Binary Robust Independent Elementary Features (BRIEF) [7], where point pairs are sampled from isotropic Gaussian distribution. Binary Robust Invariant Scalable Keypoints (BRISK) [23], in turn, uses a circular pattern with equally spaced points for this purpose, and Oriented FAST and Rotated BRIEF (ORB) [8, 10, 33] uses a learned sampling pattern and FAST [32] technique to generate keypoints. A retinal sampling pattern is used in Fast Retina Keypoint (FREAK) [2]. These approaches perform binary tests on pixel intensities, what can make them more sensitive to noise. Therefore, recently introduced binary descriptors perform tests on larger patch regions. For example, in Local Difference Binary (LDB) [39, 40] and Modified-LDB [3] (AKAZE), the described patch is divided into 4, 9, 16, and 25 disjunctive blocks of pixels (cells) and then binary tests are performed on their mean intensities and directional gradients. A more distinctive approach can be found in [30], where Binary Robust Fast Features (BRAF) descriptor employs four image patches with scale-dependent sizes are divided into 3×3 pixel blocks, or in Binary Descriptor with Shared Pixel Blocks (BDSB) [29], in which overlapping pixel regions are used. These pixel block-based descriptors perform all-against-all tests separately for each patch and for each the type of information extracted from patches, i.e., intensities and gradients.

In the literature, apart from hand-crafted solutions, there are also data-driven binary approaches in which the image patch is divided into pixel regions and then values used for their description are compared optimising performance criteria. Tests on intensities and gradients of regional invariants can be seen in Ordinal and Spatial information of Regional Invariants (OSRI) [38]. In OSRI, a resulted binary string is long and requires further reduction. This is also present in LDB [40]. BinBoost descriptor [36] replaces binary tests with learned binary hash functions [36]. AdaBoost classifier and gradient-based image features are used here. Receptive Fields Descriptor (RFD) [9] uses thresholded fields' responses of rectangular or Gaussian pooling regions, and Binary Online Learned Descriptor (BOLD) [4] is independently optimised for each image patch. The state-of-the-art learning-based binary descriptors (e.g., BinBoost), as well as hand-crafted block-based descriptors (e.g., LDB, AKAZE) have computation time close to floating-point techniques, what makes a binary descriptor build on the top of a floating-point technique still worthy of consideration. Furthermore, dimensionality reduction approaches with a projection matrix may also suffer from longer computation time due to multiplication of feature vectors by the matrix [35].

The technique introduced in this paper is related to pixel block-based binary descriptors, in which tests on values representing some image regions within the patch are performed. Since modifications of SIFT and SURF are presented and evaluated in further parts of this paper, it can be said that they are novel binary representatives of these techniques.

The rest of this paper is organised as follows. Section 2 covers the description of the proposed approach and its application. Experimental results with related discussions are presented in Section 3. Finally, Section 4 concludes the paper and indicates possible directions of future research.

2. Approach

2.1. Binary Tests for a Real-Valued Descriptor

It can be assumed that a real-valued descriptor provides a description of the image patch centred at the detected keypoint. Let B_i denotes a pixel block within the patch, $i = 1, 2 \ldots, N$, where N is the number of such blocks. Each i-th block is described with a vector V_j^i having M real-valued dimensions, $V_{j=1}^i, V_{j=2}^i, \ldots, V_{j=M}^i$. The size of the patch and its division depends on descriptor computation pipeline. In the proposed approach, the binary string is created using all-against-all binary tests on values representing blocks [3,29,30,39,40]. Thus, the computation of the binary string b can be written as:

$$b = \sum_{j=1}^{M} \left(\sum_{1 \leq o \leq {}^N C_2} 2^{o-1} T_j \right), \qquad (1)$$

where o denotes a pair of blocks, B_l and B_k, $l \neq k, \{l, k\} = 1, 2, \ldots, N$. In the equation, the sum sign denotes concatenation of strings with binary values. There are ${}^N C_2 = N! \setminus (2!(N-2)!)$ binary tests for each j-th dimension, the test T_j is calculated using eq. (2):

$$T_j = \begin{cases} 1, & if\, V_j^l < V_j^k \\ 0, & otherwise. \end{cases} \qquad (2)$$

2.2. Binary SIFT and SURF

In SIFT [24], an image patch centred at keypoint location is extracted. Its size depends on the keypoint's scale. Then, local gradients are used to provide a dominant orientation of the patch. The orientation is used for orienting local gradients determined for 4×4 grids placed within each of 16 pixel blocks. The gradients are quantised into eight angular bins, and then 128-dimensional, weighted histogram is determined. The histogram is created using magnitudes and orientations. Magnitudes are weighted using Gaussian, and the resulting vector is normalised to unit vector to ensure robustness against illumination changes. Extension of SIFT descriptor pipeline using the approach presented in this paper is applied as follows. Each of 16 blocks ($N = 16$) is represented by eight-dimensional ($M = 8$), real-valued vector. Therefore, there are 120 binary tests (${}^{16}C_2$) for each dimension, and the finally obtained binary string has 960 bits. The string is long and that may require additional dimensionality reduction (as e.g., 21576-bit string in OSRI [38]). However, in some applications it would be more convenient to use a small number of longer descriptors with high discriminative properties than a large number of worse performing descriptors with short binary strings.

In SURF [5], similarly to SIFT, a square patch is divided into 16 pixel blocks (4×4), and the descriptor is created by a union of vectors resulted from sums of horizontal and vertical Haar wavelet responses and their absolute values. The Haar wavelet responses for each pixel block are computed at 5×5 regularly spaced sample points and then weighted with a Gaussian in order to introduce better robustness against geometrical transformations. Finally, each pixel block is described using four values, then the 64-dimensional vector is further transformed into unit vector to provide contrast invariance [5]. Since the patch is described using N = 16 blocks, and each block is represented by M = 4 values, the resulting binary string is composed of 480 bits ($4 \times {}^{16}C_2$). It is worth noticing that recently introduced AKAZE has a similar length (486 bits).

3. Experimental Evaluation

The binary versions of SIFT and SURF created with the feature representation introduced in this paper were evaluated and compared with their floating-point counterparts. The state-of-the-art techniques designed to reduce their high-dimensionality (LDAHash [35], binarised SIFT [31], binarised SURF [31]) were also used, as well as the state-of-the-art binary descriptors (AKAZE [3], BRISK [23], and ORB [33]). AKAZE, being an example of hand-crafted pixel block-based descriptor, is closer to the introduced technique, in terms of performed steps in descriptor's computation pipeline, than binarisation techniques applied to the resulted floating-point vectors. All three compared binary descriptors are equipped in the full descriptor pipeline, i.e., they were designed to be able to detect and describe interest points.

In experiments, single-threaded applications with descriptors were run on a CPU with Intel Core i5-5200u 2.2 GHz, 8 GB RAM, and Microsoft Windows 7. Java implementations of SURF and SIFT from BoofCV library (http://boofcv.org/) were used [1], as well as Java interface to OpenCV implementations of AKAZE [3], BRISK [23], and ORB [33] descriptors (http://opencv.org, https://github.com/bytedeco/javacv). The author of this work implemented LDAHash and approaches introduced by Peker [31] in Java using known projection matrices (LDAHash, http://cvlab.epfl.ch/research/detect/ldahash) or the published description [31].

3.1. Image Matching

The compared techniques were evaluated in terms of the area under *Recall vs. 1-Precision* curve calculated for corresponding pairs of images that belong to image benchmarks designed to test matching performance of local feature descriptors [6,12,13,25,27]. For this purpose, Oxford [25] and Heinly *et al.* [13] datasets were used. The datasets contain base images and sequences of images that exhibit different amount of most popular transformations, e.g., rotation, scaling, viewpoint change, blur, illumination, exposure, and JPEG compression. Exemplary image sentences from these datasets can be seen on fig. 1. In order to calculate the area under *Recall vs. 1-Precision* curve, for the base image and its transformed equivalent, a predefined number of interest points on both images were detected and described. Then, for each keypoint from the base image its corresponding keypoint from the second image was determined, taking into account a distance ratio between the closest and the second closest keypoint from that image, localisation error of the found keypoint and an overlap size between described image patches. In this paper, the following values were used [13, 25]: 0.8 distance ratio, three pixel localisation error, and 40% overlap. Binary descriptors were compared using Hamming distance, while floating-point vectors were compared with Euclidean distance. *Precision* of matching was calculated as the number of returned correct matches to the all matches, and *Recall* as the number of returned correct matches to all possible correct matches [13]. Finally, *Recall vs. 1-Precision* curves were obtained using threshold-based similarity matching [5] for 500 keypoints with the strongest response per image.

Obtained results are presented in Tab. 1 and Tab. 2. On the basis of matching results it can be said that for Oxford dataset the introduced binary versions of SURF and SIFT performed slightly worse than their real-valued counterparts. However, for Heinly *et al.* dataset and the sequences from Oxford dataset with illumination and blur changes, they were better. Furthermore, SURF^b and SIFT^b clearly outperformed other feature binarisation techniques. Here, SIFT^b was the leading binary technique for seven image sequences, and six times was the best descriptor in general. SURF^b, in turn, outperformed other techniques six times, and for two image sequences was the leading performer.

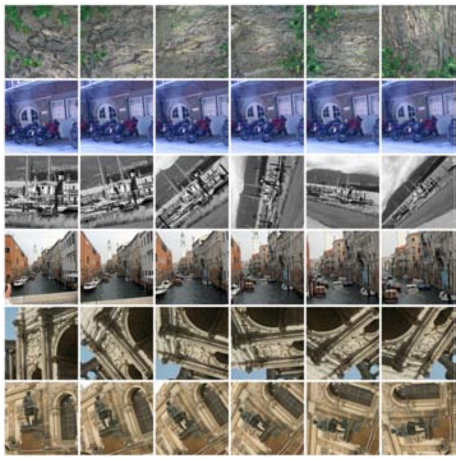


Fig. 1. Exemplary images from Oxford and Heinly et al. banchmark datasets

The comparison of description time of compared approaches, as well as their lengths, is given in tab. 4. In the table, the average timings measured per keypoint for *Bikes* sequence from Oxford dataset [25] are reported. It can be seen that the presented approach requires the addition of 2 to 6% of descriptor computation time in order to obtain binary representation of described image patch. This takes longer than simple thresholding [31], but is two times faster than more advanced LDAHash. Binary descriptors designed to perform fast detection and description of interest points are faster, but as it was shown in image matching experiments, as well as in recognition tests (Section 3.2), the usage of the introduced technique is justified by its superior performance. Such performance is often important in practical tasks, where unequivocal description of an image patch may lead to, e.g., near-duplicate content detection [20] The matching time depends on the length of the descriptor, and both descriptors are shorter than their floating-point counterparts. SIFT^b's binary string allows for faster matching than 128-dimensional vector used by SIFT. However, a practical usage of both techniques may require an application of a salient bit-selection technique [29, 38, 40].

3.2. Recognition

The performance of descriptors was also compared in matching-based recognition tests on two demanding image collections,

Tab. 1. Comparison of matching performance measured in terms of mean area under Recall vs. 1- Precision curves on Oxford dataset

Method	Image sequence and transformation							
	Bark Rotation	*Bikes* Blur	*Boat* Rotation	*Graffiti* Viewpoint	*Leuven* Illumination	*Ubc* JPEG	*Wall* Viewpoint	*Trees* Blur
SURF [5]	**0.278**	0.578	**0.386**	0.180	0.587	**0.746**	0.438	0.382
SIFT [24]	0.144	0.534	0.212	0.177	0.642	0.604	**0.454**	**0.391**
LDAHash [35]	0.118	0.557	0.176	0.157	0.624	0.587	0.363	0.291
Bin. SURF [31]	0.045	0.400	0.113	0.047	0.408	0.496	0.105	0.105
Bin. SIFT [31]	0.103	0.341	0.110	0.074	0.485	0.496	0.322	0.237
AKAZE [3]	0.177	0.368	0.199	0.121	0.383	0.466	0.245	0.204
BRISK [23]	0.057	0.324	0.035	0.064	0.132	0.333	0.091	0.129
ORB [33]	0.033	0.060	0.043	0.025	0.133	0.181	0.090	0.077
SURF^b	<u>0.229</u>	<u>**0.627**</u>	<u>0.347</u>	0.164	0.619	<u>0.733</u>	0.323	<u>0.325</u>
SIFT^b	0.124	0.615	0.199	<u>**0.225**</u>	<u>**0.675**</u>	0.600	<u>0.380</u>	0.286

Note: The best value for each image sequence is written in boldface, the best result for the binary approach is underlined.

Tab. 2. Comparison of matching performance measured in terms of mean area under Recall vs. 1- Precision curves on Heinly et al. dataset

Method	Image sequence and transformation				
	Ceiling Rotation	*Day and night* Illumination	*Rome* Rotation	*Semper* Rotation	*Venice* Scaling
SURF [5]	0.458	0.061	0.563	0.318	0.652
SIFT [24]	0.616	0.158	0.658	0.62	0.151
LDAHash [35]	0.604	0.156	0.665	0.61	0.138
Bin. SURF [31]	0.257	0.037	0.358	0.239	0.326
Bin. SIFT [31]	0.542	0.057	0.407	0.549	0.120
AKAZE [3]	0.464	0.038	0.354	0.267	0.225
BRISK [23]	0.320	0.050	0.355	0.395	0.244
ORB [33]	0.083	0.019	0.125	0.119	0.135
SURF^b	0.439	0.128	0.603	0.387	<u>**0.664**</u>
SIFT^b	<u>**0.625**</u>	<u>**0.199**</u>	<u>**0.690**</u>	<u>**0.633**</u>	0.164

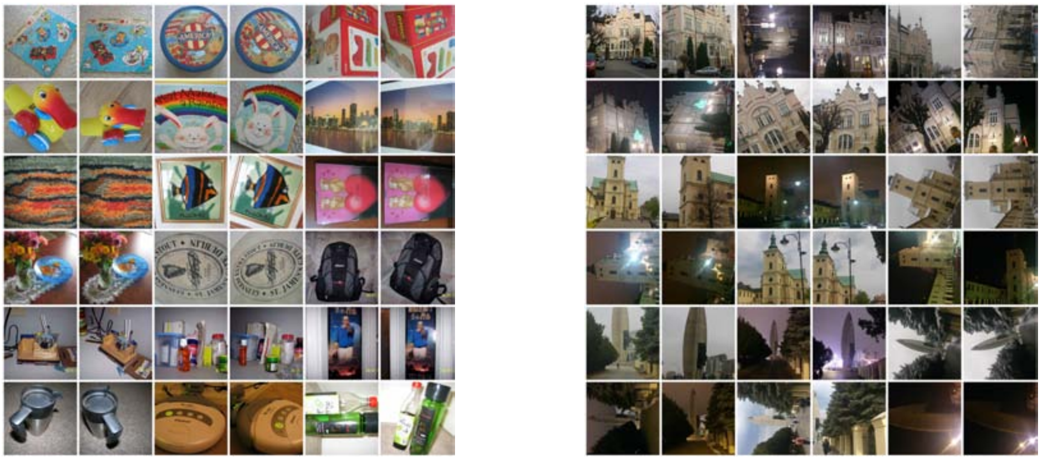
Note: The best value for each image sequence is written in boldface, the best result for the binary approach is underlined.

Tab. 3. Comparison of recognition performance on the BR and UKBench datasets

Method	Test set in the BR dataset						UKBench
	Autumn day	*Autumn night*	*Winter day*	*Winter night*	*Spring day*	*Spring night*	
	Recognition rate (in %)						Score
SURF [5]	34.6	40.6	32.2	47.4	40.4	41.8	2.444
SIFT [24]	56.8	45.4	62.4	53.6	65.0	50.0	3.000
LDAHash [35]	42.6	37.8	51.2	47.2	54.8	44.4	2.859
Bin. SURF [31]	28.0	31.4	23.0	32.0	25.4	32.8	2.262
Bin. SIFT [31]	23.4	18.0	30.0	19.0	23.0	22.0	2.342
AKAZE [3]	32.2	40.8	28.8	44.0	40.6	58.0	2.757
BRISK [23]	2.60	2.40	4.40	2.40	4.00	8.60	1.347
ORB [33]	16.2	16.0	11.6	16.0	18.0	36.4	2.199
SURF^b	52.2	**68.4**	47.2	**65.2**	54.0	**66.8**	2.791
SIFT^b	**58.0**	54.6	**72.0**	60.6	**65.8**	52.8	**3.071**

Note: The best value for each test is written in boldface.

the UKBench [28] and the Beautiful Rzeszow (http://marosz.kia.prz.edu.pl/br.html) [30] datasets. Some images from these datasets are shown on fig. 2. The UKBench dataset contains images of 2550 objects. There are four images per object. According to the proposed performance index [28], an average of top four results for the first objects' images is used. The second benchmark used in recognition tests, the Beautiful Rzeszow (BR) dataset, contains 3000 images of 50 sites in Rzeszow, Poland. There are 10 images of



(a) (b)

Fig. 2. Exemplary images from object recognition benchmarks: (a) UKBench, and (b) the BR

Tab. 4. Comparison of description time and length

Method	Description time (ms)	Length
SURF [5]	0.140	64 Floats
SIFT [24]	0.752	128 Bytes
LDAHash [35]	SIFT + 38 us	128 Bits
Bin. SURF [31]	SURF + 2 us	64 Bits
Bin. SIFT [31]	SIFT + 3 us	128 Bits
AKAZE [3]	0.141	486 Bits
BRISK [23]	0.042	512 Bits
ORB [33]	0.022	256 Bits
SURF^b	SURF + 9 us	480 Bits
SIFT^b	SIFT + 20 us	960 Bits

Note: Description time is given per keypoint.

each site captured at a different time of the day (day and night) and season (spring, autumn, and winter), in 2015. The objects were photographed introducing viewpoint, scale, and rotation changes. The time the images were captured also resulted in challenging illumination conditions and many occlusions, which are not present in popular UKBench. The recognition performance on this dataset was tested using images taken at the given time of the day as queries and the top one returned results was assessed. Here, the recognition accuracy is presented as the percent of correctly recognised objects. In the object recognition tests, test images were recognised using k-nearest neighbour classifier (k=1), which assigned learned labels taking into account the largest number of returned matched descriptor pairs between the test and the learning images. In matching, the threshold of 0.8 was used. For UKBench, due to long matching time for real-valued descriptors, the score was reported on the basis of the first 3000 images.

Table 3 contains recognition results for these two datasets. It can be seen that both introduced binary descriptors, SURF^b and SIFT^b, were better than compared techniques. Due to high robustness of SIFT descriptor, tests with day images were easier for its derivative approaches (SIFT^b and LDAHash). However, for the night images, SURF^b turned out to be better, since performed binary tests increased illumination invariance of the keypoint's description. Interestingly, both techniques improved results obtained by SURF and SIFT, what with faster matching time and shorter descriptor length has a practical importance. This can be also observed for UKBench dataset, where SIFT^b outperformed other methods. For this dataset, SIFT was better than SURF, what also indicates that the developed binary representation is limited by the distinctive properties of binarised real-valued descriptor.

4. Conclusion

In this paper, an approach to an extension of floating-point descriptor's computation pipeline with a step that allows turning them into binary descriptors was shown. The approach performed binary tests on values determined by floating-point descriptors for pixel blocks within the image patch in order to create a binary string. The resulted binary representation was evaluated and compared with popular feature binarisation techniques, as well as with state-of-the-art binary descriptors equipped with full descriptor pipeline. Obtained results on four image benchmarks are promising and showed that two introduced binary representations built on top of SURF and SIFT techniques are highly competitive, outperforming other approaches in matching and recognition tasks.

Future works will consider dimensionality reduction of obtained binary strings, as well as experiments with other real-valued, local descriptors, or other descriptor types [37].

AUTHOR

Mariusz Oszust – Department of Computer and Control Engineering, Rzeszow University of Technology, Wincentego Pola 2, 35-959 Rzeszow, Poland, e-mail: marosz@kia.prz.edu.pl, www: www.marosz.kia.prz.edu.pl.

REFERENCES

[1] P. Abeles, "Speeding up SURF". In: *Proc. Int. Symp. on Advances in Visual Computing (ISVC)*, 2013, 454–464.

[2] A. Alahi, R. Ortiz, and P. Vandergheynst, "FREAK: Fast retina keypoint". In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, 2012, 510–517.

[3] P. F. Alcantarilla, J. Nuevo, and A. Bartoli, "Fast explicit diffusion for accelerated features in nonlinear scale spaces". In: *British Machine Vision Conf. (BMVC)*, 2013.

[4] V. Balntas, L. Tang, and K. Mikolajczyk, "BOLD - Binary online learned descriptor for efficient image matching". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, 2367–2375.

[5] H. Bay, T. Tuytelaars, and L. V. Gool. "SURF: Speeded up robust features". In: *Proc. European Conf. on Computer Vision (ECCV)*, 404–417. Springer, May 2006.

[6] S. Bianco, D. Mazzini, D. P. Pau, and R. Schettini, "Local detectors and compact descriptors for visual search: A quantitative comparison", *Digit. Signal Process.*, vol. 44, 2015, 1–13.

[7] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. "BRIEF: Binary Robust Independent Elementary Features". In: K. Daniilidis, P. Maragos, and N. Paragios, eds., *Computer Vision - ECCV 2010*, volume 6314 of *Lecture Notes in Computer Science*, 778–792. Springer Berlin Heidelberg, 2010.

[8] C. Clarke and P. Angelov, "Sariva: Smartphone app for real-time intelligent video analytics", *Journal of Automation, Mobile Robotics and Intelligent Systems*, vol. 8, no. 4, 2014, 15–19.

[9] B. Fan, Q. Kong, T. Trzcinski, Z. Wang, C. Pan, and P. Fua, "Receptive fields selection for binary feature description", *IEEE T. Image Process.*, vol. 23, no. 6, 2014, 2583–2595.

[10] J. Figat, T. Kornuta, and W. Kasprzak, "Performance evaluation of binary descriptors of local features". In: L. Chmielewski, R. Kozera, B.-S. Shin, and K. Wojciechowski, eds., *Proceedings of the International Conference on Computer Vision and Graphics*, vol. 8671, 2014, 187–194.

[11] J. Fuentes-Pacheco, J. Ruiz-Ascencio, and J. M. Rendon-Mancha, "Visual simultaneous localization and mapping: A survey", *Artif. Intell. Rev.*, vol. 43, no. 1, 2015, 55–81.

[12] E. Garcia-Fidalgo and A. Ortiz, "Vision-based topological mapping and localization methods: A survey", *Robotics and Autonomous Systems*, vol. 64, 2015, 1–20.

[13] J. Heinly, E. Dunn, and J.-M. Frahm. "Comparative evaluation of binary features". In: *Proc. European Conf. on Computer Vision (ECCV)*, 759–773. Springer, Oct. 2012.

[14] A. Hietanen, J. Lankinen, J.-K. Kamarainen, A. G. Buch, and N. Kruger, "A comparison of feature detectors and descriptors for object class matching", *Neurocomputing*, vol. 184, 2016, 3 – 12, RoLoD: Robust Local Descriptors for Computer Vision 2014.

[15] W. Hu, N. Xie, L. Li, X. Zeng, and S. Maybank, "A survey on visual content-based video indexing and retrieval", *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 41, no. 6, 2011, 797–819.

[16] G. Hua, M. Brown, and S. Winder, "Discriminant embedding for local image descriptors". In: *2007 IEEE 11th International Conference on Computer Vision*, 2007, 1–8.

[17] W. Jan, "On the representation of planes for efficient graph-based slam with high-level features", *Journal of Automation, Mobile Robotics and Intelligent Systems*, vol. 10, no. 03, 2016, 3–11.

[18] Z. Ji, Y. Pang, and X. Li, "Relevance preserving projection and ranking for web image search reranking", *IEEE Trans. Image Process.*, vol. 24, no. 11, 2015, 4137–4147.

[19] Y. Ke and R. Sukthankar, "PCA-SIFT: a more distinctive representation for local image descriptors". In: *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Conference on*, vol. 2, 2004, II–506–II–513 Vol.2.

[20] Y. Ke, R. Sukthankar, L. Huston, Y. Ke, and R. Sukthankar, "Efficient near-duplicate detection and sub-image retrieval". In: *ACM Multimedia*, vol. 4, no. 1, 2004, 5.

[21] N. Khan, B. McCane, and S. Mills, "Better than SIFT?", *Mach. Vis. Appl.*, vol. 26, no. 6, 2015, 819–836.

[22] S. Kim, K. Paeng, J. W. Seo, and S. D. Kim, "Bi-DCT: DCT-based local binary descriptor for dense stereo matching", *IEEE Signal Proc. Let.*, vol. 22, no. 7, 2015, 847–851.

[23] S. Leutenegger, M. Chli, and R. Siegwart, "BRISK: Binary Robust Invariant Scalable Keypoints". In: *Computer Vision (ICCV), 2011 IEEE International Conference on*, 2011, 2548–2555.

[24] D. G. Lowe, "Distinctive image features from scale-invariant keypoints", *Int. J. Comput. Vision*, vol. 60, no. 2, 2004, 91–110.

[25] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, 2005, 1615–1630.

[26] O. Miksik and K. Mikolajczyk, "Evaluation of local detectors and descriptors for fast feature matching". In: *Proc. Int. Conf. on Pattern Recognition (ICPR)*, 2012, 2681–2684.

[27] D. Mukherjee, Q. M. J. Wu, and G. Wang, "A comparative experimental study of image feature detectors and descriptors", *Mach. Vision Appl.*, vol. 26, no. 4, 2015, 443–466.

[28] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree". In: *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 2, 2006, 2161–2168.

[29] M. Oszust, "BDSB: Binary descriptor with shared pixel blocks", *Journal of Visual Communication and Image Representation*, vol. 41, 2016, 154–165.

[30] M. Oszust, "Towards binary robust fast features using the comparison of pixel blocks", *Meas. Sci. Technol.*, vol. 27, no. 3, 2016, 035402.

[31] K. A. Peker, "Binary SIFT: Fast image retrieval using binary quantized SIFT features". In: *Content-Based Multimedia Indexing (CBMI), 2011 9th International Workshop on*, 2011, 217–222.

[32] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, 2010, 105–119.

[33] 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*, 2011, 2564–2571.

[34] A. Smeulders, D. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, 2014, 1442–1468.

[35] C. Strecha, A. Bronstein, M. Bronstein, and P. Fua, "LDAHash: Improved matching with smaller descriptors", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 1, 2012.

[36] T. Trzcinski, M. Christoudias, P. Fua, and V. Lepetit, "Boosting binary keypoint descriptors". In: *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, 2013, 2874–2881.

[37] D. Warchol and M. Wysocki, "Recognition of hand posture based on a point cloud descriptor and a feature of extended fingers", *Journal of Automation, Mobile Robotics and Intelligent Systems*, vol. 10, no. 01, 2016, 48–57.

[38] X. Xu, L. Tian, J. Feng, and J. Zhou, "OSRI: A rotationally invariant binary descriptor", *IEEE Trans. Image Process.*, vol. 23, no. 7, 2014, 2983–2995.

[39] X. Yang and K. T. T. Cheng, "LDB: An ultra-fast feature for scalable augmented reality on mobile devices". In: *Mixed and Augmented Reality (ISMAR), 2012 IEEE International Symposium on*, 2012, 49–57.

[40] X. Yang and K. T. T. Cheng, "Local difference binary for ultrafast and distinctive feature description", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 1, 2014, 188–194.