Performance Evaluation of Binary Descriptors of Local Features

Jan Figat, Tomasz Kornuta, and Włodzimierz Kasprzak

Warsaw University of Technology, Institute of Control and Computation Eng.
Nowowiejska 15/19 00-665 Warsaw, Poland
{J.Figat,T.Kornuta,W.Kasprzak}@ia.pw.edu.pl

Abstract. The article is devoted to the evaluation of performance of image features with binary descriptors for the purpose of their utilization in recognition of objects by service robots. In the conducted experiments we used the dataset and followed the methodology proposed by Mikolajczyk and Schmid. The performance analysis takes into account the discriminative power of a combination of keypoint detector and feature descriptor, as well as time consumption.

Keywords: performance evaluation, image features, binary descriptors, SIFT, FAST, BRIEF, BRISK, ORB, FREAK.

1 Motivation of the Work

The recently observed progress in service robotics would not be possible without the progress in the recognition of everyday objects. Our two-handed robot Velma posses an active head equipped with a pair of RGB cameras and a pair of vertically mounted Kinect sensors. Such a perception subsystem enables the robot to acquire point clouds constituting its environment. We developed a process of generation of 3D object models consisting of two types of point clouds: a dense colour point cloud (used mainly for visualisation) and a sparse feature cloud (used for recognition). Currently our object recognition process relies on SIFT (Scale Invariant Feature Transform) [1] features transformed into a feature cloud on the basis of additional depth information. We have chosen SIFT because it is one of the most valued feature.

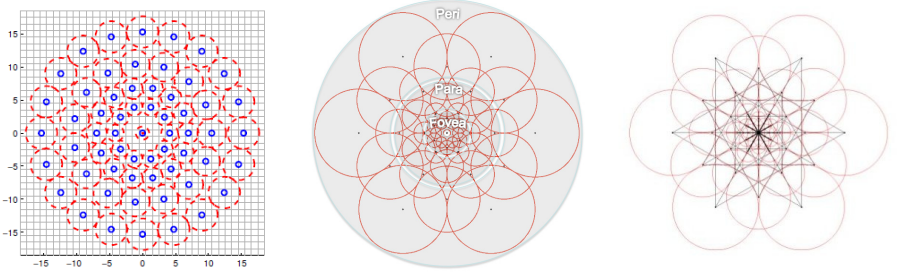
However, the recent advent in image features turned our attention to features possessing binary descriptors. The advantage of utilization of those type of features is simple: reduced time consumption. This is achieved due to the fact that instead of computation of all gradients for each pixel in the patch, binary descriptors are encoded on the basis of comparison of intensity of pairs of selected pixels. Additionally, the comparison between the binary descriptors is much faster from the classical HOG-like descriptors because it bases on the Hamming distance, which can be computed by summing the bits being result of the XOR operation between the two compared binary strings. Hence utilization of such a feature in the process of a real-time recognition of objects by a service

L.J. Chmielewski et al. (Eds.): ICCVG 2014, LNCS 8671, pp. 187–194, 2014.

robot is highly desirable. With several types of features currently present in order to select the one that fits best to our needs we examined their properties and compared their discriminative power. In this paper we present the results of such a performance evaluation.

2 Local Features with Binary Descriptors

Typically, a local feature consists of a detector (which detects stable keypoints) and a descriptor (characterising its neighbourhood). BRIEF (Binary Robust Independent Elementary Features) [2] offers a binary descriptor, without the proposal of its own method of detection of keypoints, thus typically it is combined with FAST (Features from Accelerated Segment Test) [3] detector. The BRIEF descriptor usually contains 128, 256 or 512 bits whereas the size is equal to the number of analysed pairs of pixels of analysed patch. Hence the number influences both the speed rate and discriminative power. The descriptor is sensitive to noise, because for each pixel of a given pair it considers only the point intensity, disregarding the neighbouring pixels. This sensitivity can be reduced by prior smoothing of the image and typically Gaussian filter is used. BRIEF does not have a constant sampling pattern. Instead pairs of pixels used for building of the descriptor are randomly selected. The authors proposed five methods of determination of the point pairs and pointed that the best results were achieved with the use of random selection with Gaussian distribution.



(a) BRISK (source: [4]) (b) FREAK (source: [5]) (c) FREAK orientation pairs (source: [5])

Fig. 1. Sampling patterns

ORB (Oriented FAST and Rotated BRIEF) [6] is similar to BRIEF with added rotation and scaling invariance. Besides, instead of using a randomly selected pairs, ORB learns the optimal set of sampling pairs using machine learning techniques. ORB uses FAST to find keypoints. Additionally, it builds image pyramid to achieve scale robustness. The rotation invariance is obtained by using moments, which are computed in a circular-shaped patch around the center

of its mass. Sampling pairs should have two properties: they should be uncorrelated and the chosen set of pairs should be characterized with possibly maximal variance (it will make the feature more discriminative). To fulfil those needs, ORB runs a greedy search among all pairs following a predefined binary tests.

The BRISK (Binary Robust Invariant Scalable Keypoints) [4] descriptor is using a hand-crafted sampling pattern, composed out of concentric rings, with more points on outer rings. Fig. 1a presents BRISK sampling pattern with 60 sampling points. The small blue circles represent the sampling points locations, whereas the radiuses of red dashed circles are correspond to the standard deviation of the Gaussian kernel used to smooth the intensity values at the sampling points. Two types of sampling-point pairs are distinguished: short-distance and long-distance ones. Authors proposed to set the thresholds of distances depending on the scale in which keypoint was detected. BRISK also possess an orientation compensation mechanism.

The FREAK (Fast Retina Keypoint) [5] descriptor, similarly to BRISK, uses an encoded sampling pattern (fig. 1b). This pattern uses overlapping concentric circles with more points on inner rings. Each circle represents a sensitive field. In order to achieve the rotation invariance, FREAK samples pairs with symmetric sensitive fields with respect to the patch center, as shown on fig. 1c. FREAK is also similar to ORB by learning the optimal set of sampling pairs. First it creates a set of pairs mimicking the saccadic search (human retina movements) and subsequently uses machine learning to select subset possessing the most discriminative power.

3 Performance Evaluation

3.1 Dataset

Our methodology of evaluation of performance of image features follows the work of Mikolajczyk and Schmid [7] and, besides others, we decided to use their image dataset. The dataset used contains images subjected to six different distortions (image transformations), namely: blurring, change of viewpoint (rotation), change of scale, JPEG compression, change in illumination. It is divided into eight images subsets, as presented in fig. 2. Those subsets are named bikes, trees, graffiti, wall, bark, boat, lueven and UBC respectively. Each of such a subsets consists of six images: one considered as basic image and five being more and more distorted. Additionally, the distorted images are supplemented with files containing homography between the basic image and considered one.

3.2 Performance Evaluation

Fig. 3 presents the developed process for evaluation of performance of image features. For each image of a given pair (containing basic and distorted images) we first detect keypoints with a given detector and subsequently extract the associated descriptors.

Fig. 2. Dataset used in the performance evaluation [7]

Next features from those two sets are compared in order to find the best matches. The knowledge of the proper homography between the two analysed images enables us to transform the positions of features extracted from the distorted image into the equivalent position in basic image. We treat this as a ground truth and reject all correspondences with difference in image positions being grater than a given parameter. We checked the results for distance being equal to 1, 2, 3, 4 and 5 pixels and noticed that the most optimal results were obtained for the distance equal to 2, so during further experiments set the distance to 2.

During the experiments we also measured the time of keypoint detection, descriptor extractor and feature matching. In our implementation we used the OpenCV [8] library (version 2.4.8) running on a PC with a quadcore Phenom II 965 processor and 4GB RAM, under control of Ubuntu 12.0.4 OS.

3.3 Results of Experiments

First set of tests was performed for all of the abovedescribed binary descriptors, basing on exactly the same set of keypoints. Because our goal was to find a combination of detector and descriptor giving better (or at least not behaving much worse) then the featured currently used in our tasks, we applied the SIFT detector for localization of keypoints and measured the SIFT descriptor performance, treating it as a reference. Results of comparison of the percentage of correctly found correspondences with keypoints localized by SIFT detector are presented inin fig. 4a. In this case SIFT performance simply dominates binary descriptors.

Next, we decided to conduct the same experiments for default detectors (using FAST of those features that do not have their own, special detectors i.e. BRIEF and FREAK). Fig. 4b presents results of such a comparison. We can observe that the best results were obtained once again for SIFT, however in several cases ORB acted almost as good, and sometimes even better.

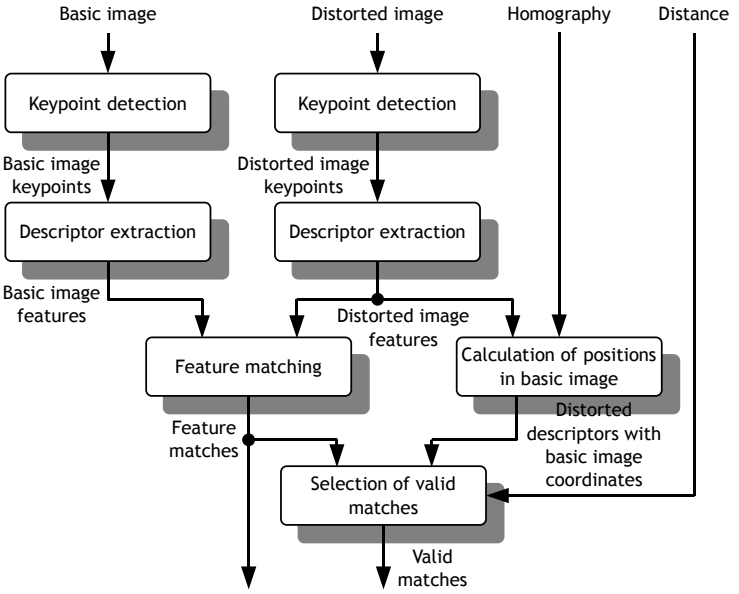
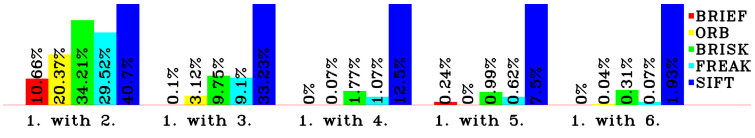
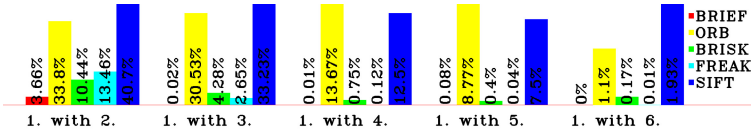


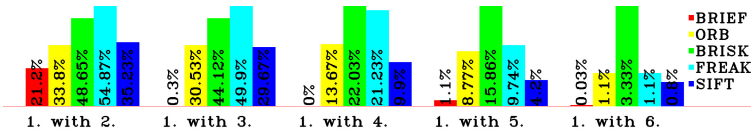
Fig. 3. Process of evaluation of features



(a) SIFT detector

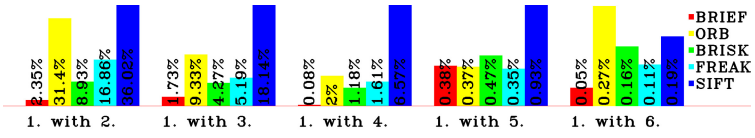


(b) default detector

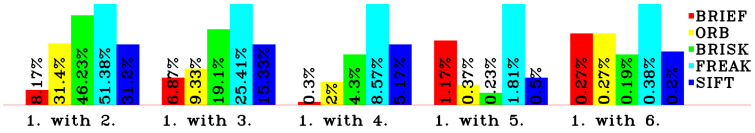


(c) ORB detector

Fig. 4. Percentage of correctly determined matches (boat subset)



(a) default detector



(b) ORB detector

Fig. 5. Percentage of correctly determined matches (graffiti subset)

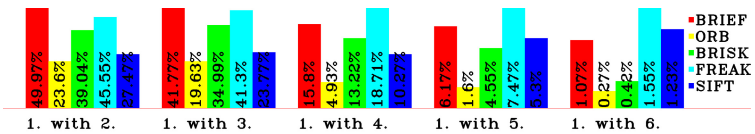
Finally, we evaluated the performance for all descriptors on keypoints detected by the ORB detector (fig. 4c). In this case both ORB and SIFT descriptors were overwhelmed with BRISK and FREAK, both acting in the majority cases even better then SIFT for SIFT-detected keypoints.

Similar results were obtained for both datasets dealing with changes of a zoom with additional rotation (boat, bark), as well as for viewpoint changes (wall, graffiti). In particular, for graffiti images the best results were obtained for ORB detector with FREAK, beating all other combinations (fig. 5b). In case of the wall subset the combination of ORB detector with FREAK descriptor also appeared to be one of the best (fig. 6a).

It is worth noting that for other image subsets the results of performation evaluation were not that unanimous. However, as it was mentioned earlier, we are seeking features for a given purpose, i.e. recognition of indoor objects, hence robustness against blurring or image compression is not so important to us.

The detection time per detected keypoint was presented in the tab. 1. As it shows the FAST detector is the fastest and SIFT detector is the slowest. Besides, it is important to note that ORB detector is almost twice time faster then BRISK but almost 20 times slower then FAST.

In the tab. 1 the time feature extraction per detected feature was shown. As we can see the extraction time for SIFT descriptor was far more time-consuming then for binary descriptors. The longest feature extraction time for the binary descriptors was for ORB, but still it was more then ten times faster then for SIFT.



(a) wall subset

Fig. 6. Percentage of correctly determined matches for ORB detector

Table 1. Average feature detection and extraction times (boat images)

Detector	Detection time [μs]
FAST	0.87857
ORB	16.39273
BRISK	30.63530
SIFT	229.67313

(a) Average keypoint detection time

Descriptor	Extraction time [μs]
BRIEF	7,42990
ORB	14.5747
BRISK	9.60987
FREAK	9.09653
SIFT	174.44800

(b) Average descriptor extraction time

Table 2. Average detection and extraction time for boat images

Detector	Descriptor	Average time per feature point [μs]
FAST	BRIEF	8.30847
ORB	ORB	30.96741
BRISK	BRISK	39.30988
ORB	BRISK	27.95955
FAST	FREAK	11.33364
ORB	FREAK	32.42584
SIFT	SIFT	404.12113

The decision to choose an appropriate descriptor with detector was based on both the time of operation as well as the needs of our research. From the tab. 2 it can be seen that FREAK with ORB detector is a little bit slower than BRISK with ORB detector, but more than ten times faster than the SIFT with SIFT detector.

3.4 Conclusions

The results obtained for ORB detector with FREAK descriptor were much better than for others descriptors, especially for the viewpoint changes. Surprisingly, these results were event better then for classical SIFT detector and descriptor combination. For the zoom with rotation changes, combination of ORB detector with FREAK descriptor seemed to be a little bit worse than ORB wit BRISK, whereas for the point of view changes the results were much better than the performance of the other descriptors. Additionally, in comparison to SIFT, the time consumption for combination of ORB with FREAK is one order of magnitude smaller (tab. 2). Therefore, we decided that a combination of ORB detector with FREAK descriptor fits best to our needs.

4 Summary

The article was devoted to the evaluation of performance of local features with binary descriptors. We evaluated the features with binary descriptors, taking

SIFT as ground truth. Comparison was made for several combinations of detectors and descriptors. Aside of the discriminative performance of features we analysed the time consumption for various combinations of keypoint detectors and extraction of descriptors. As a result we have chosen ORB detector and FREAK descriptor, which seem to be the best for the purpose of recognition of everyday objects in the indoor environment.

In our future work will plan to use the selected combination in the object recognition and generation of 3D models of objects. Aside of that, during the experiments it appeared that the chosen dataset does not entirely fulfil our needs. In particular, distortions such as blur or JPEG compression are not important for service robots performing manipulation tasks, but instead systematic studies of rotation (viewpoint change), scaling, occlusions and object damages (due to e.g. scratches resulting from repeated grasping of objects with a robot gripper) would be required. The last one we find especially interesting.

Acknowledgments. The authors acknowledge the support of European Union within the RAPP project funded by the 7th Framework Programme (Collaborative Project FP7-ICT 610947). The authors would also like to thank to Łukasz Jendrzejek and Karol Koniuszewski for help with the implementation and initial experiments.

References

1. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)
2. Calonder, M., Lepetit, V., Strecha, C., Fua, P.: BRIEF: Binary Robust Independent Elementary Features. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 778–792. Springer, Heidelberg (2010)
3. Rosten, E., Drummond, T.: Machine learning for high-speed corner detection. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006, Part I. LNCS, vol. 3951, pp. 430–443. Springer, Heidelberg (2006)
4. Leutenegger, S., Chli, M., Siegwart, R.Y.: Brisk: Binary robust invariant scalable keypoints. In: 2011 IEEE International Conference on Computer Vision (ICCV), pp. 2548–2555. IEEE (2011)
5. Alahi, A., Ortiz, R., Vandergheynst, P.: FREAK: Fast Retina Keypoint. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 510–517. IEEE (2012)
6. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: Orb: an efficient alternative to sift or surf. In: 2011 IEEE International Conference on Computer Vision (ICCV), pp. 2564–2571. IEEE (2011)
7. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(10), 1615–1630 (2005)
8. Bradski, G., Kaehler, A.: Learning OpenCV: Computer Vision with the OpenCV Library, 1st edn. O'Reilly (September 2008)