Comparative Assessment of Point Feature Detectors and Descriptors in the Context of Robot Navigation

Received 10th October 2012; accepted 22nd November 2012.

Adam Schmidt, Marek Kraft, Michał Fularz, Zuzanna Domagała

Abstract:
This paper presents evaluation of various contemporary interest point detector and descriptor pairs in the context of robot navigation. The robustness of the detectors and descriptors is assessed using publicly available datasets: the first gathered from the camera mounted on the industrial robot [17] and the second gathered from the mobile robot [20]. The most efficient detectors and descriptors for the visual robot navigation are selected.

Keywords: *point features, detectors, descriptors*

1. Introduction

The detection, description and matching of point features plays a vital role in most of the contemporary algorithms for visual odometry [1, 2] or visual simultaneous localization and mapping [3, 4]. Over the last decade several new fast detectors such as FAST (Features from Accelerated Segment Test) [5], SURF (Speeded Up Robust Features) [6], CenSurE (Center Surround Extrema) being basis for STAR [7] and descriptors e.g. SURF [6], BRIEF (Binary Robust Independent Features) [8], ORB (Oriented FAST and Rotated Brief) [9], FREAK (Fast Retina Keypoint) [10] have been proposed and successfully applied to robot navigation tasks. As the processing speed is the key aspect in such tasks, some of the detectors and descriptors have been either implemented in FPGA (Field Programmable Gate Array) [11, 12] or simplified [13].

At the moment, to the extent of authors knowledge there is no comparative study of the newest point detectors and descriptors with regard to their applicability in robot navigation. In [14] and [15] the authors describe the desired characteristics of point detectors and descriptors, however they do not present any experimental results. The authors in [17] have compared various interest point detectors using sequences recorded with the camera placed on an industrial robot. In the follow up research they compared the detector-descriptor pairs efficiency, however only cross-correlation, SIFT (Scale Invariant Feature Transform) [19] and DAISY descriptors were considered. Another experimental study has been presented in [16] where the detector-descriptor pairs have been graded according to the number of feature matches supporting the 8-point algorithm solution found by the RANSAC algorithm. This indirect evaluation method was forced by difficulties in gathering the ground truth correspondence data for image pairs.

This paper presents an evaluation of detector-descriptor pairs in the context of robot navigation. The measure of the pair's efficiency is based on the reprojection error of point feature pairs matched on two images. The images used were selected from the publicly available datasets [17, 20]. The analysis allows selecting detector and descriptor pair most suitable for application in the robot navigation both in the context of accuracy.

The rest of the paper is structured as follows: Section 2 provides a description of the detectors and descriptors evaluated in the study. Procedure is presented in Section 3. Section 4 contains results while Section 5 contains concluding remarks and the planned future work.

2. Detectors and descriptors

2.1. The Shi-Tomasi feature detector

The Shi-Tomasi (GFTT) feature detector is relies on investigating the local auto-correlation function of the image intensity function [21]. To perform this, the so called structural tensor is used (Eq. 1).

$$A = \sum_u \sum_v w(p,q) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \tag{1}$$

The I_x and I_y terms denote the partial image derivatives in x and y directions, respectively. The term $w(p,q)$ denotes the convolution with a weighting window over the area (p,q). A Gaussian is the most common choice for the weighting window, as it makes the responses isotropic. The feature strength measure M_c is then defined as given in equation 2.

$$M_c = min(|\lambda_1|, |\lambda_2|) \tag{2}$$

Corners are the local maxima of M_c above some arbitrary threshold t.

2.2. FAST feature detector

The FAST [5] feature detector inspects the values of the intensity function of pixels on a circle of the radius r around the candidate point p. The pixel on a circle is considered 'bright' if its intensity value is brighter by at least t, and 'dark' if its intensity value is darker by at least t than the intensity value of p, where t is some arbitrary threshold. The candidate pixel is classified as a feature on a basis of a segment test – if a contiguous, at least n pixels long arc of 'bright' or 'dark' pixels is found on the circle. The original solution uses $r = 3$ and $n = 9$. An illustration of the segment test is given in Fig. 1.

The ID3 algorithm is used to optimize the order in which pixels are tested, resulting in high computational efficiency. The segment test alone produces small sets of adjacent positive responses. To further refine the results, an additional cornerness measure is used for non-maximum suppression (NMS). As the NMS is applied to only a fraction of image points that successfully passed the segment test, the processing time is kept short.

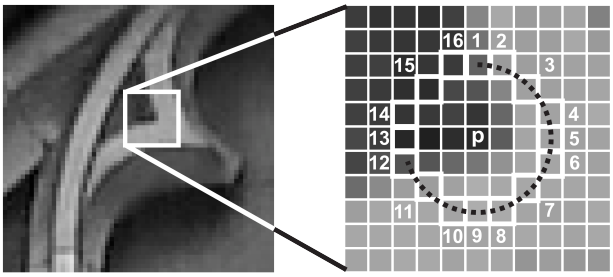

Fig. 1. The illustration of the segment test used by the FAST feature detector.

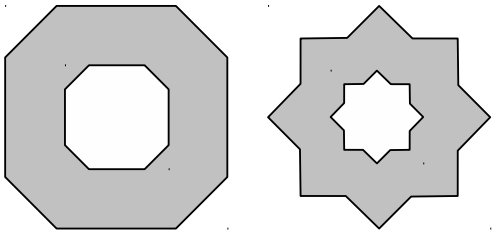

Fig. 2. Masks used by the CenSurE (left) and STAR (right) feature descriptors.

2.3. SURF feature detector

SURF [6] is an image feature detector and descriptor, inspired by the SIFT detector/descriptor. The main motivation for the development of SURF has been to overcome SIFT's main weakness – its computational complexity and low execution speed. SURF has been reported to be up to a few times faster than SIFT without compromising the performance. The detection step in SURF takes advantage of the Haar wavelet approximation of the blob detector based on the Hessian determinant. The approximations of Haar wavelets can be efficiently computed using integral images, regardless of the scale. Accurate localization of multiscale SURF features requires interpolation.

2.4. STAR feature detector

The STAR keypoint detector has been developed as a part of the OpenCV computer vision library. It is a derivative of the CenSurE (Center Surround Extrema) feature detector [7]. The authors of the solution aimed at the creation of a multiscale detector with full spatial resolution. As described in [7], the subsampling performed by SIFT and SURF affects the accuracy of feature localization. The detector uses a bi-level approximation of the Laplacian of Gaussians (LoG) filter. The circular shape of the mask is replaced by an approximation that allows to preserve rotational invariance and enables the use of integral images for efficient computation. Scale-space is constructed without interpolation, by applying masks of different size. The shape of the masks used for feature detection by CenSurE and STAR are given in Fig. 2.

2.5. SURF feature descriptor

The SURF [6] feature descriptor uses Haar wavelets in conjunction with integral images to encode the distribution of pixel intensity values in the neighborhood of the detected feature while accounting for the feature's scale. Computation of the descriptor for a given feature at the scale s begins with the assignment of the dominant orientation to make the descriptor rotation invariant. The process starts with computing the Haar wavelet responses in two dominant directions for every point within the radius of $6s$ from the feature.

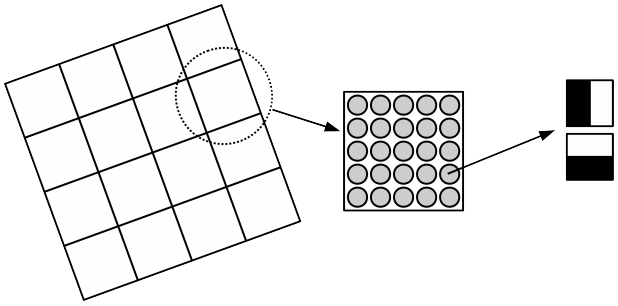

Fig. 3. Division of the processed window into subregions and sampling points as used by the SURF feature descriptor.

The size of the square wavelet masks is also adjusted according to the feature scale and set to $4s$. The responses are then weighted with a Gaussian centered at the feature point. Each one of the responses gives rise to a point in the vector space, with the x-responses along the abscissa and the y-responses along the ordinate. Afterwards, a circle segment covering an angle of $\frac{\pi}{3}$ is rotated around the origin (feature point). The responses under the segment are summed and form a resultant vector. The rotation angle corresponding to the longest resultant vector is selected as the dominant orientation of the feature descriptor.

The computation of the descriptor itself starts with placing a square window with a side length of $20s$ centered on the feature point and oriented as computed in the previous step. The window is divided into 4×4 regular square subregions. Each subregion is divided into 5×5 uniformly distributed sample points. For each sample point, the Haar wavelet responses for two principal directions are computed as shown in Fig. 3.

Each subregion contributes to the descriptor with four components: the sums of the responses in the two principal directions (d_x, d_y) and their absolute values, as given in equation 3.

$$DESC_{sub} = [\sum dx, \sum dy, \sum |dx|, \sum |dy|] \quad (3)$$

The responses from the 16 subregions are once again weighted with a Gaussian mask according to the relative position of the subregion and the analyzed point. For 16 subregions, the descriptor size is 64.

2.6. BRIEF feature descriptor

The BRIEF [8] descriptor proposed in [8] uses binary strings for feature description and subsequent matching. This enables the use of Hamming distance to compute the descriptors similarity. Such similarity measure can be computed very efficiently – much faster than the commonly used L_2 norm. Due to BRIEF's sensitivity to noise, the image is smoothed with a simple averaging filter before applying the actual descriptor. The value of each bit contributing to the descriptor depends on the result of a comparison between the intensity values of two points inside an image segment centered on the currently described feature.

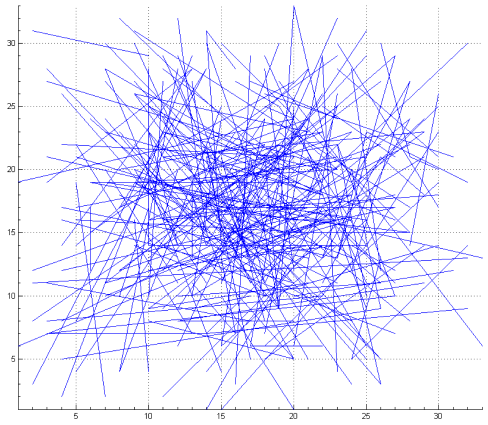

Fig. 4. An example random sampling pattern used by the BRIEF feature descriptor; point pairs used for comparison are connected with lines.

The bit corresponding to a given point pair is set to 1 if the intensity value of the first point of this pair is higher than the intensity value of the second point, and reset otherwise. The authors of the original solution tested a few sampling strategies for selecting the point pairs. The empirical results have shown that sampling according to the Gaussian distribution centered on the described feature point results in best performance. The proposed descriptor is 512-bit long and computed over a 48×48 pixel image patch, although the length of the descriptor and the size of the window can be changed to adapt to the application at hand. The initial smoothing is performed with a 9×9 pixel rectangular averaging filter. The basic form of BRIEF is not invariant w.r.t. rotation. Example random sampling pattern used by BRIEF is given in Fig. 4.

2.7. ORB feature descriptor

The ORB [9] (Oriented FAST and Rotated BRIEF) descriptor extends the BRIEF descriptor by adding two important improvements. The first one is to augment the descriptor with orientation data from the FAST feature detector. This allows to make the descriptor robust to in-plane rotation. This is done by rotating the coordinates of the point pairs for binary tests around the described feature by the feature orientation angle. Second innovation is the selection scheme for point pairs whose comparisons contribute to the descriptor. The random sampling has been replaced with a sampling scheme that uses machine learning for de-correlating BRIEF features under rotational invariance. This makes the nearest neighbor search during matching less error-prone.

2.8. FREAK feature descriptor

The FREAK [10] (Fast Retina Keypoint) descriptor is another extension of the basic concepts of BRIEF [8]. It provides the descriptor with feature orientation by summing the estimated local gradients over selected point pairs. Using a specific point sampling pattern allows to apply more coarse discretization of rotation, allowing for savings in memory consumption. A special, biologically inspired sampling pattern is also used. While the resulting descriptor is still a binary string, the sampling pattern allows for the use of a 'coarse-to-fine' approach to feature description.

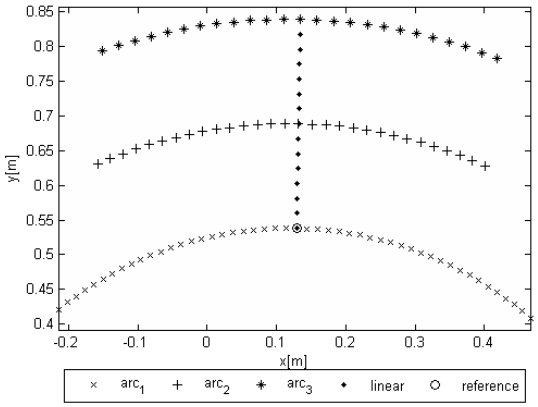

Fig. 5. The four trajectories of the Robot Data Set and the position of the reference frame.

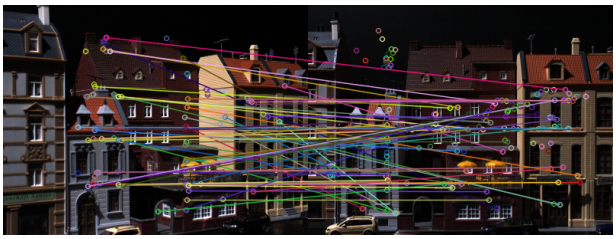

Fig. 6. Exemplary pair of images from the Robot Data Set with matches using the FAST detector and BRIEF descriptor.

Point pairs carrying the information on most distinctive characteristics of the feature neighborhood are compared in the first place. This allows for faster rejection of false matches and shortening of the computation time.

3. Experiments

3.1. Datasets

Two datasets were used in the experiments. The Robot Data Set [17, 18] consists of 60 scenes, registered from 119 positions under varying lighting conditions using ahigh resolution camera mounted on the industrial robot. The camera positions form 4 trajectories: 3 angular with constant distance from the scene and one linear with constant camera heading (Fig. 5). Such a diverse dataset allows to evaluate the robustness of detector-descriptor pairs with regard to the scale, rotation and illumination changes. Exemplary images from the dataset are presented in the Fig. 6.

The second dataset used in the experiment is a video sequence gathered with the Kinect sensor mounted on the wheeled robot [20]. The 'Pioneer SLAM' sequence consisting of 2921 images was used. The robot's trajectory during the sequence registration is presented on the Fig. 7. Exemplary images from the second dataset are presented in the Fig. 8.

The above datasets have been selected for the experiment as they reflect the two scenarios encountered in the context of mobile robot navigation. The first represents the situation in which the mobile robot observes the same scene from a number of positions differing in distance and viewing angle. It is important to notice that the camera re-

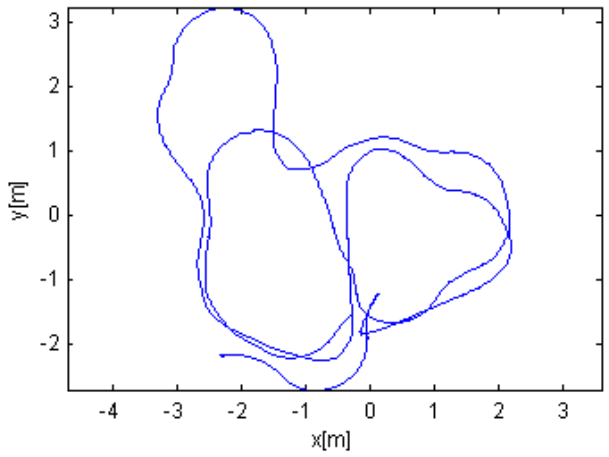

Fig. 7. The 'Pioneer SLAM' trajectory.

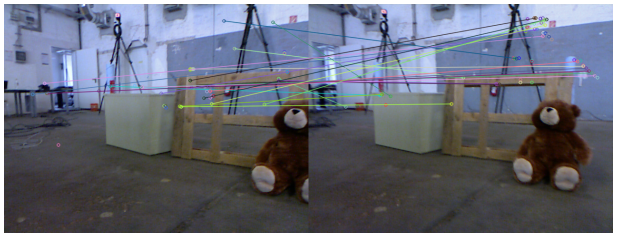

Fig. 8. Exemplary images from the Freiburg Data Set with matches using the FAST detector and BRIEF descriptor.

mains on a constant altitude which would also be the case for most of the mobile robots. The ability to match feature points and consequently recognize a previously seen scene is crucial especially in the context of loop-closing.

The second dataset presents the challenges usually present in the tasks used in the localization of the mobile robot. The sequence consists of images gathered from the onboard camera where the consecutive images differ only slightly and usually no in-plane rotation is present. The ability to detect and match features in such sequences is especially important in tasks such as the visual odometry or visual SLAM.

Both the datasets contain images, ground truth data of the camera position, camera intrinsic parameters and distortion coefficients.

3.2. Evaluation

The following procedure was performed for every analyzed pair of images:

1) the point features were detected on both images using the selected detector;
2) the point features descriptors were calculated using the selected descriptor algorithm;
3) the coordinates of the features were undistorted and normalized according to the camera parameters;
4) the essential matrix E describing the images' epipolar geometry was calculated using the relative ground truth translation ($t = \left[\begin{array}{ccc} t_x & t_y & t_z \end{array} \right]^T$) and rotation ($R$) between the two camera positions:

$$E = R \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}; \tag{4}$$

5) the features from both images were matched by minimizing the distance between their descriptors resulting in the set of quadruples (u_i, v_i, U_j, V_j) where (u_i, v_i) are the normalized coordinates of the feature extracted from the first image and (U_j, V_j) are the normalized coordinates of the best matching feature from the second image;
6) for each quadruple the symmetric reprojection error was calculated according to:

$$err = MAX\left(\left|e_i, (U_j, V_j)\right|, \left|e_j, (u_i, v_i)\right|\right) \tag{5}$$

where e_i and e_j are the epipolar lines defined as:

$$\begin{aligned} a_i x + b_i y + c_i &= 0 && (6)\\ A_j x + B_j y + C_j &= 0 && (7)\\ \left[\begin{array}{ccc} a_i & b_i & c_i \end{array} \right]^T &= E \left[\begin{array}{ccc} u_i, v_i, 1 \end{array} \right]^T && (8)\\ \left[\begin{array}{ccc} A_j & B_j & C_j \end{array} \right]^T &= \left[\begin{array}{ccc} U_j, V_j, 1 \end{array} \right] E && (9) \end{aligned}$$

If the error was smaller than the threshold $thresh$ the match was considered an inlier;

7) the ratio of the number of inliers to the number of matches was calculated.

The final score of the detector-descriptor pair over a dataset was calculated as the mean of the inliers to matches ratios of all the image pairs in the dataset.

4. Results

The Table 1 presents the average number of features found by the evaluated detectors on the images from both datasets. It is clearly visible that the STAR detector has been able to find the smallest number of features – over six times fewer than the second-worst detector.

The Tables 2–5 contain the maximal and minimal matching ratios of the tested detector-descriptor pairs over the sequences from the Robot Data Set. As expected the maximal matching ratios have been observed for the smallest angular or linear displacement. Increasing the displacement has caused the reduction of the matching ratio down to the minimal values that have been observed for the biggest displacements. Please confront Fig. 9–14.

Analogously, the Table 6 presents the maximal and minimal matching ratios of the detector desciptor pairs over the Freiburg dataset. The smallest ratios have been observed for the biggest difference in frames. Suprisingly, the best matching has been observed while matching every-second frame. This shows that the computation of the visual odometry parameters under too small displacement of features between two consecutive images is inaccurate (Fig. 15).

The experiments show, that binary vector based BRIEF and ORB descriptors have clearly outperform the other tested solutions. Interestingly, the third binary vector descriptor – FREAK – displayed a significantly lower matching accuracy than the other tested solutions, contrary to the claims in [10]. The popular SURF feature descriptor performed relatively well, but since it offers lower matching accuracy than BRIEF and ORB and is slower to compute and match, there are strong arguments against using it.

Tab. 1. The average number of detected features.

Detector	Robot Image Dataset	Freiburg Dataset
FAST	1469	634
Pyramid FAST	2292	1074
GFTT	1469	868
PyramidGFTT	2984	773
SURF	4580	1421
StarKeypoint	221	101

SURF is also the descriptor being most influenced by the type of the detector it is paired with. This is visible especially in the results from the Freiburg dataset (figure 15). It is worth noting that the use of pyramid FAST and GFTT detectors with a multi-scale SURF and ORB descriptor does not increase the matching performance when using a scale-aware descriptor. This is rather surprising, and is probably caused by the necessity of performing interpolation to determine the location of the feature, which negatively impacts on the feature location accuracy in higher scales.

The Star keypoint detector is not burdened by this additional inaccuracy, as it offers full location accuracy across all scales. The results for the linear sequence shown in figure 14 support this claim. It must be noted however, that the Star descriptor requires a relatively feature-rich environment, as the average feature count it returns is relatively low as shown in Table 1.

5. Conclusions

The various, contemporary point features detector and descriptor pairs were compared in order to determine the best combination for the task of robot visual navigation. The sequences chosen as a testbed displayed typical point feature distortions encountered during indoor mobile robot navigation – scaling and affine transformation with very little or none in-plane rotation.

As all descriptors perform well when paired with the FAST corner detector, the FAST-BRIEF pair is a good choice when processing speed is a concern. Under the camera movement conditions featured in both of the test sequences used, the additional computational cost to bear when using descriptors and detectors robust to in-plane rotation and large scaling seems to be unjustified. This also confirms the need for testing the detector-descriptor pairs in the context of the application, as the requirements raised by it may differ significantly from the requirements raised by typical, commonly used benchmarking image sequences. In the future it is planned to compare the efficiency of the detector-descriptor pairs in the monocular SLAM system.

AUTHORS

Adam Schmidt* – Poznań University of Technology, Institute of Control and Information Engineering, ul. Piotrowo 3A, 60-965 Poznań, Poland, e-mail: Adam.Schmidt@put.poznan.pl.

Marek Kraft – Poznań University of Technology, Institute of Control and Information Engineering, ul. Piotrowo 3A, 60-965 Poznań, Poland, e-mail: Marek.Kraft@put.poznan.pl.

Michał Fularz – Poznań University of Technology, Institute of Control and Information Engineering, ul. Piotrowo 3A, 60-965 Poznań, Poland, e-mail: Michal.Fularz@put.poznan.pl.

Zuzanna Domagała – Poznań University of Technology, Institute of Control and Information Engineering, ul. Piotrowo 3A, 60-965 Poznań, Poland, e-mail: Zuzanna.Domagala@cie.put.poznan.pl.

*Corresponding author

Acknowledgements

Adam Schmidt, Marek Kraft and Michał Fularz are scholarship holders within the project ”Scholarship support for PH.D. students specializing in majors strategic for Wielkopolska’s development“, Sub-measure 8.2.2 Human Capital Operational Programme, co-financed by European Union under the European Social Fund.

This research was financed by the National Science Centre grant funded according to the decision DEC-2011/01/N/ST7/05940, which is gratefully acknowledged.

References

[1] D. Scaramuzza, F. Fraundorfer, “Visual Odometry: Part I – The First 30 Years and Fundamentals”, *IEEE Robotics and Automation Magazine*, vol. 18(4), 2011, pp. 80–92.

[2] F. Fraundorfer, D. Scaramuzza, “Visual Odometry: Part II – Matching, Robustness and Applications”, *IEEE Robotics and Automation Magazine*, vol. 19(2), 2012, pp. 78–90.

[3] A. J. Davison, I. Reid, N. Molton and O. Stasse, “MonoSLAM: Real-Time Single Camera SLAM”, *IEEE Trans. PAMI*, vol. 29(6), 2007, pp. 1052–1067.

[4] A. Schmidt, A. Kasiński, “The Visual SLAM System for a Hexapod Robot”, *Lecture Notes in Computer Science*, vol. 6375, 2010, pp. 260–267.

[5] E. Rosten, T. Drummond, “Machine learning for high-speed corner detection”. In: *Proc. of European Conf. on Computer Vision*, 2006, pp. 430–443.

[6] H. Bay, A. Ess, T. Tuytelaars, L. Van Gool, “SURF: Speeded Up Robust Features”, *Computer Vision and Image Understanding*, vol. 110(3), 2008, pp. 346–359.

[7] M. Agrawal, K. Konolige, M.R. Blas, “CenSurE: Center surround extremas for realtime feature detection and matching”, *Lecture Notes in Computer Science*, vol. 5305, 2008, pp. 102–115.

[8] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, “BRIEF: Binary Robust Independent Elementary Features”. In: *Proceedings of ECCV*, 2010, pp. 778–792.

[9] E. Rublee, V. Rabaud, K. Konolige, G. R. Bradski, “ORB: An efficient alternative to SIFT or SURF”. In: *Proc. ICCV*, 2011, pp. 2564–2571.

[10] A. Alahi, R. Ortiz, P. Vandergheynst, “FREAK: Fast Retina Keypoint”. In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

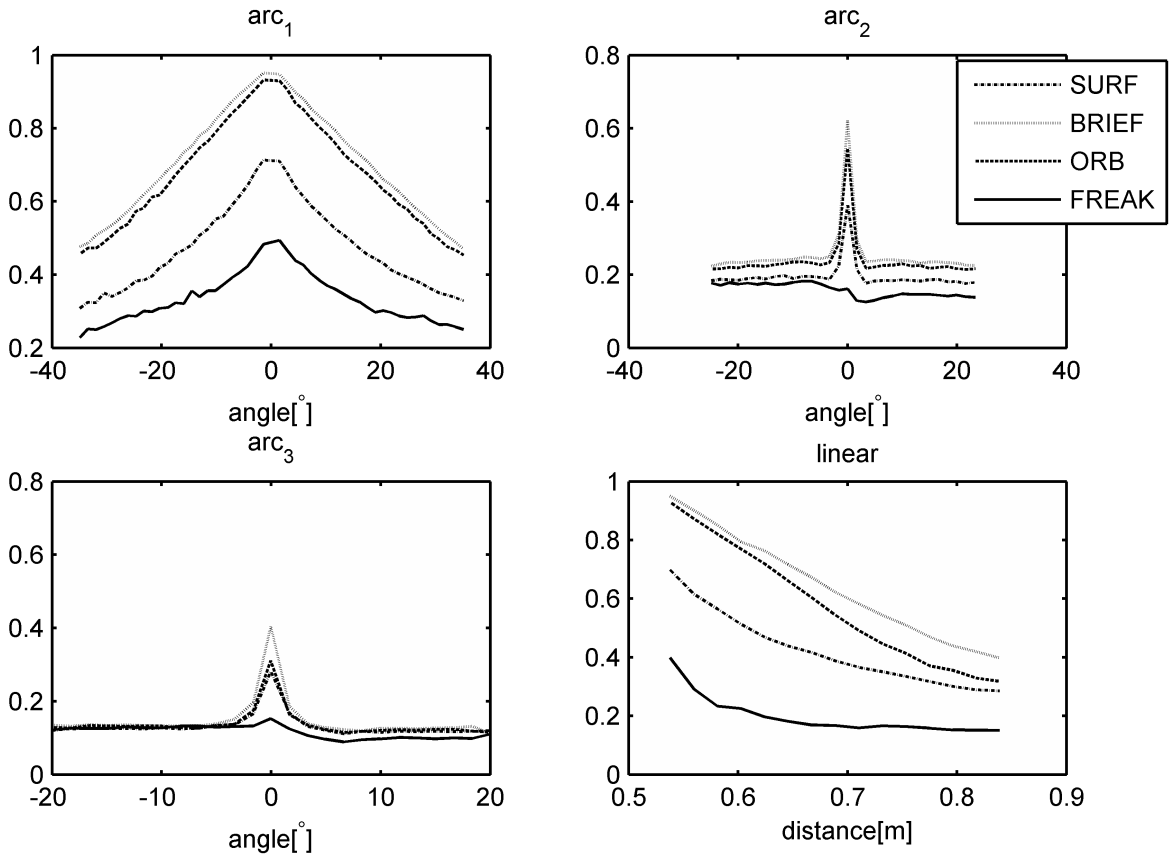

Fig. 9. Matching ratio of points detected with the FAST detector. Robot Image Dataset.

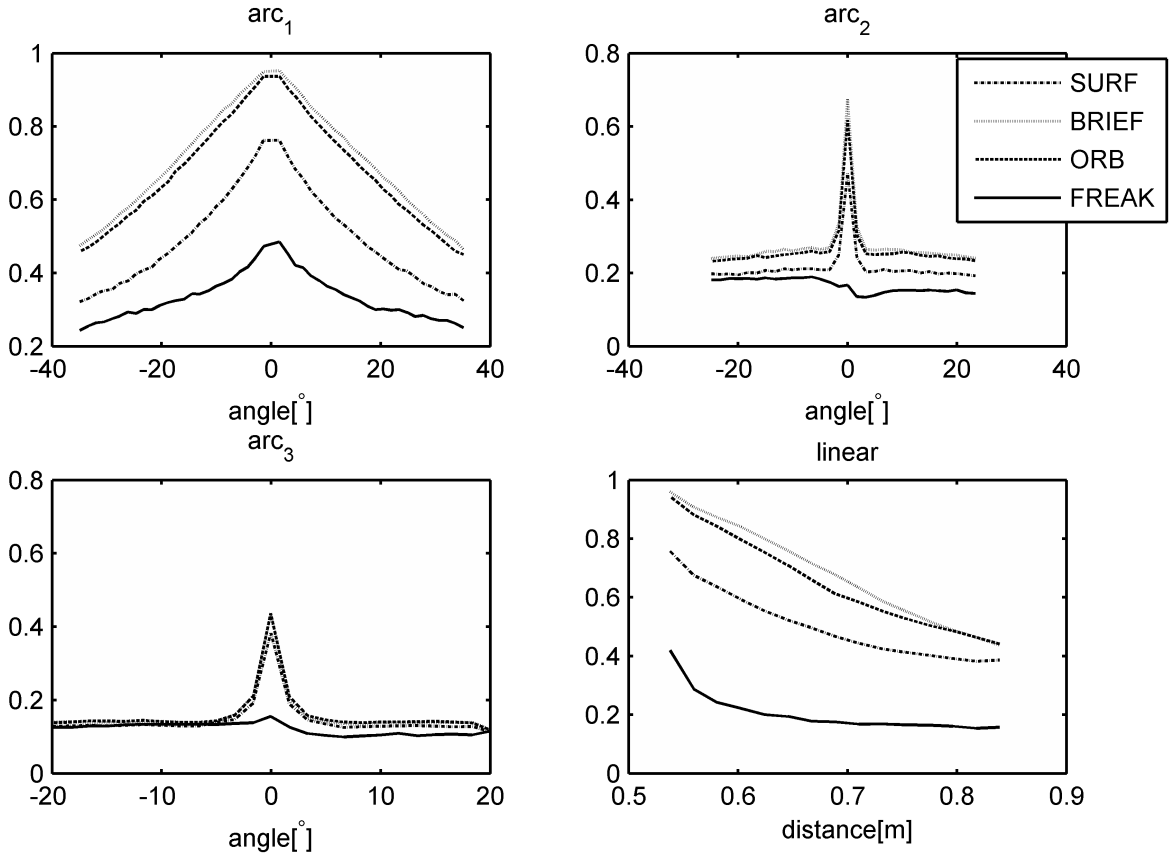

Fig. 10. Matching ratio of points detected with the Pyramid FAST detector. Robot Image Dataset.

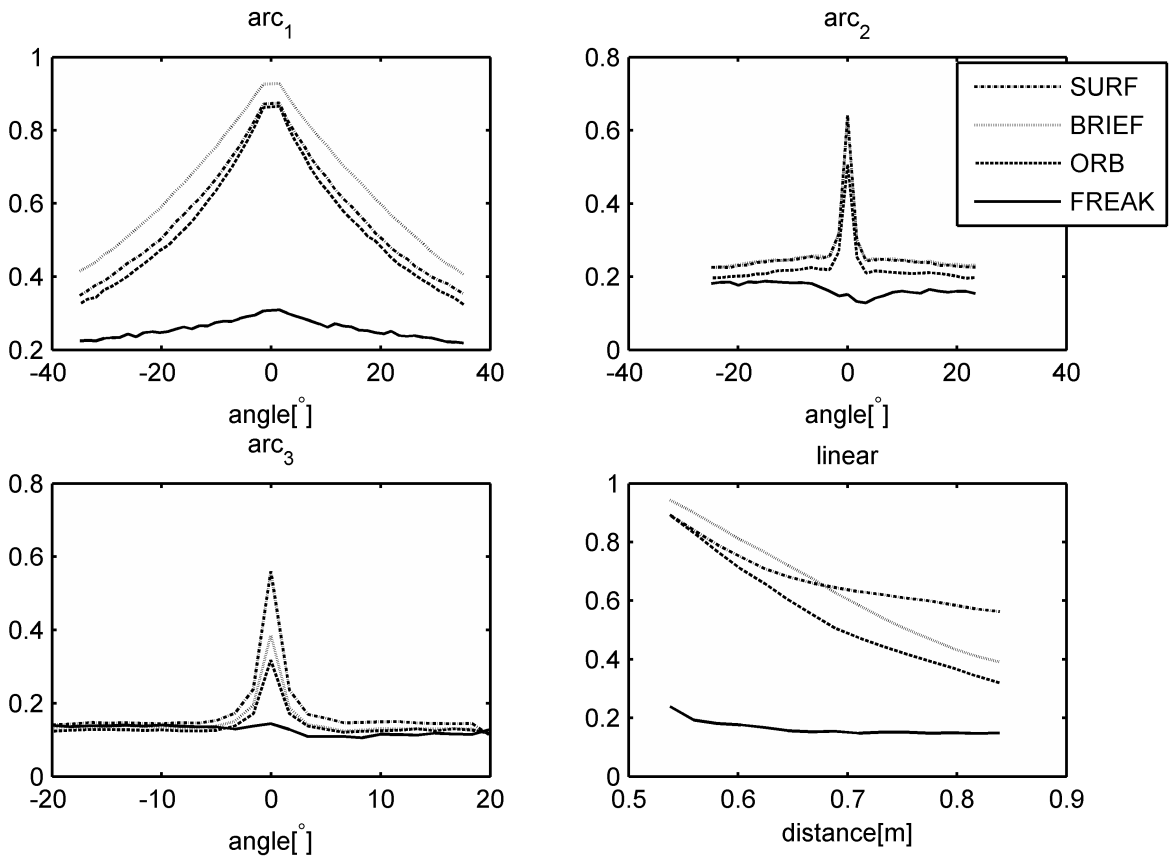

Fig. 11. Matching ratio of points detected with the GFTT detector. Robot Image Dataset.

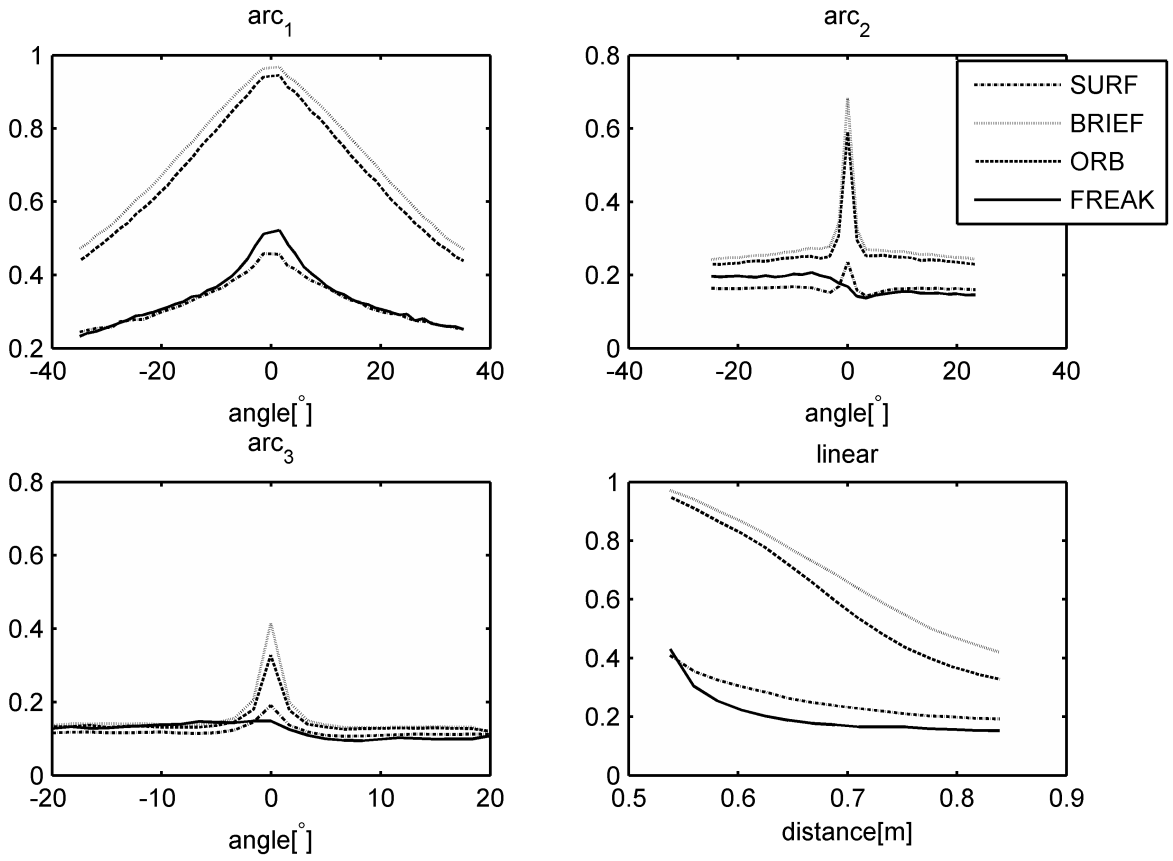

Fig. 12. Matching ratio of points detected with the Pyramid GFTT detector. Robot Image Dataset.

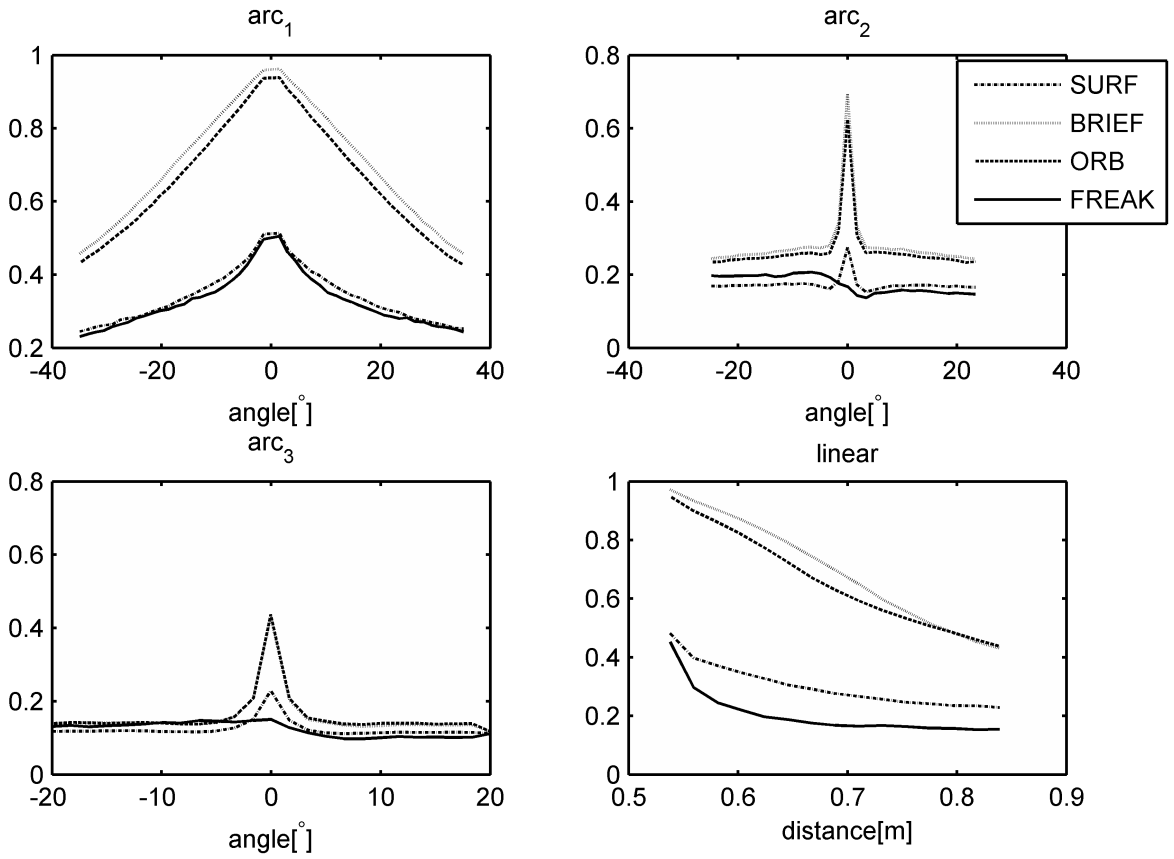

Fig. 13. Matching ratio of points detected with the SURF detector. Robot Image Dataset.

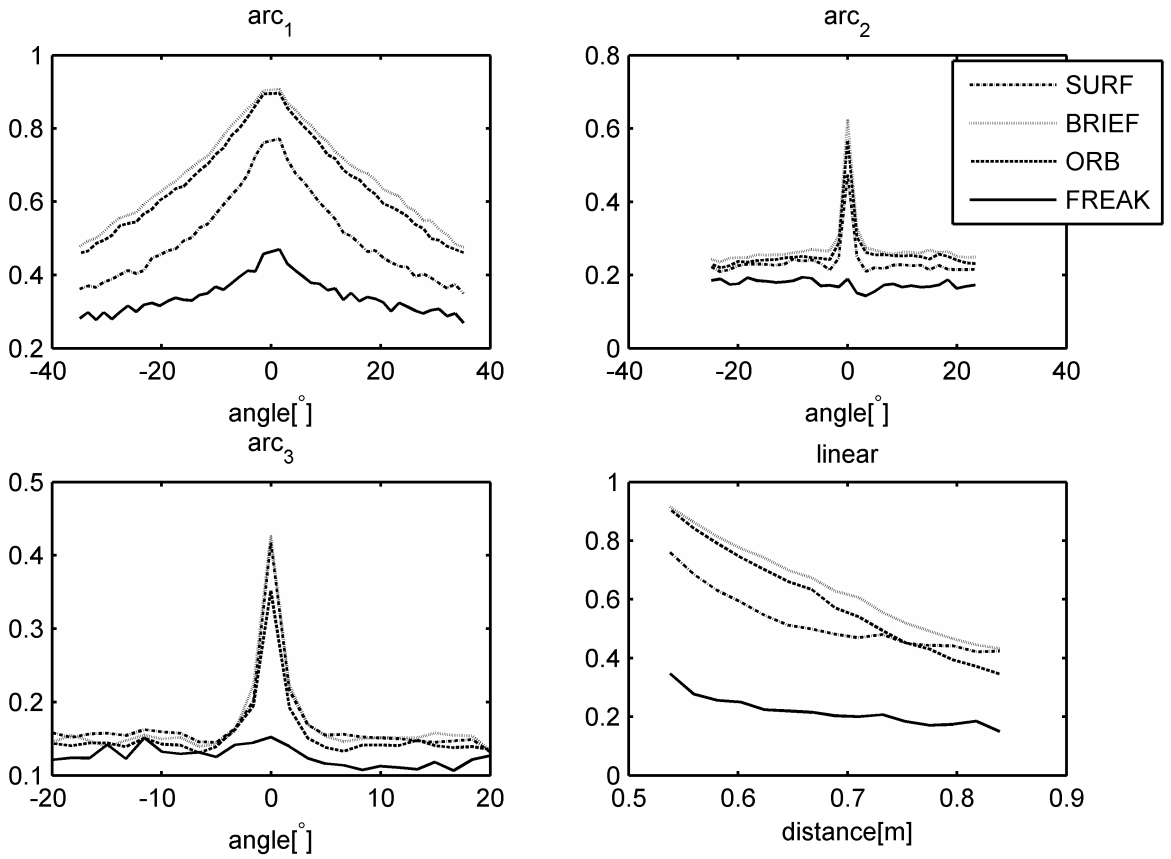

Fig. 14. Matching ratio of points detected with the StarKeypoint detector. Robot Image Dataset.

Tab. 2. Maximal and minimal matching ration over the arc_1 of the Robot Data Set.

	SURF		BRIEF		ORB		FREAK	
	MAX	MIN	MAX	MIN	MAX	MIN	MAX	MIN
FAST	0.71	0.31	0.95	0.47	0.93	0.45	0.49	0.23
PyramidFAST	0.76	0.32	0.95	0.46	0.94	0.45	0.49	0.24
GFTT	0.87	0.35	0.93	0.41	0.87	0.32	0.31	0.22
PyramidGFTT	0.46	0.24	0.97	0.47	0.95	0.44	0.52	0.23
SURF	0.51	0.24	0.96	0.46	0.94	0.43	0.51	0.23
STAR	0.77	0.35	0.91	0.48	0.90	0.46	0.47	0.27

Tab. 3. Maximal and minimal matching ration over the arc_2 of the Robot Data Set.

	SURF		BRIEF		ORB		FREAK	
	MAX	MIN	MAX	MIN	MAX	MIN	MAX	MIN
FAST	0.39	0.18	0.62	0.22	0.55	0.21	0.18	0.13
PyramidFAST	0.47	0.19	0.68	0.24	0.62	0.23	0.19	0.13
GFTT	0.64	0.23	0.63	0.23	0.51	0.20	0.19	0.13
PyramidGFTT	0.23	0.14	0.68	0.24	0.59	0.23	0.21	0.14
SURF	0.27	0.15	0.70	0.24	0.63	0.23	0.21	0.14
STAR	0.47	0.21	0.63	0.24	0.57	0.22	0.19	0.14

Tab. 4. Maximal and minimal matching ration over the arc_3 of the Robot Data Set.

	SURF		BRIEF		ORB		FREAK	
	MAX	MIN	MAX	MIN	MAX	MIN	MAX	MIN
FAST	0.28	0.11	0.40	0.11	0.31	0.12	0.15	0.09
PyramidFAST	0.38	0.11	0.43	0.12	0.44	0.12	0.16	0.10
GFTT	0.56	0.11	0.39	0.12	0.32	0.12	0.14	0.11
PyramidGFTT	0.19	0.11	0.42	0.12	0.33	0.12	0.15	0.09
SURF	0.23	0.11	0.43	0.12	0.44	0.12	0.15	0.10
STAR	0.42	0.13	0.43	0.13	0.35	0.13	0.15	0.11

Tab. 5. Maximal and minimal matching ration over the $linear$ of the Robot Data Set.

	SURF		BRIEF		ORB		FREAK	
	MAX	MIN	MAX	MIN	MAX	MIN	MAX	MIN
FAST	0.70	0.29	0.95	0.40	0.93	0.32	0.40	0.15
PyramidFAST	0.76	0.38	0.96	0.44	0.94	0.44	0.42	0.15
GFTT	0.89	0.56	0.94	0.39	0.89	0.32	0.24	0.15
PyramidGFTT	0.41	0.19	0.97	0.42	0.95	0.33	0.43	0.15
SURF	0.48	0.23	0.97	0.43	0.95	0.44	0.45	0.15
STAR	0.76	0.42	0.92	0.43	0.91	0.35	0.35	0.15

Tab. 6. Maximal and minimal matching ration over the Freiburg dataset.

	SURF		BRIEF		ORB		FREAK	
	MAX	MIN	MAX	MIN	MAX	MIN	MAX	MIN
FAST	0.42	0.24	0.76	0.39	0.73	0.38	0.19	0.14
PyramidFAST	0.53	0.28	0.78	0.40	0.76	0.39	0.20	0.15
GFTT	0.66	0.35	0.74	0.39	0.68	0.35	0.16	0.14
PyramidGFTT	0.23	0.15	0.77	0.39	0.73	0.38	0.19	0.14
SURF	0.32	0.19	0.77	0.40	0.75	0.39	0.18	0.14
STAR	0.56	0.31	0.75	0.41	0.74	0.40	0.25	0.19

[11] M. Kraft, A. Schmidt, A. Kasinski, "High-speed image feature detection using FPGA implementation of FAST algorithm". In: *Proc. VISAPP*, 2008, pp. 174–179.

[12] M. Kraft, M. Fularz, A. Kasiński, "System on chip coprocessors for high speed image feature detection and matching". In: *Proc. of Advances Concepts for Intelligent Vision Systems*, 2011, pp. 599–610.

[13] M. Kraft, A. Schmidt, "Simplifying SURF feature descriptor to achieve real-time performance". In: *Proc. Computer Recognition Systems*, 2011, pp. 431–440.

[14] Ó. Martínez, A. Gil, M. Ballesta, O. Reinoso, "Interest Point Detectors for Visual SLAM". In: *Proc. of the*

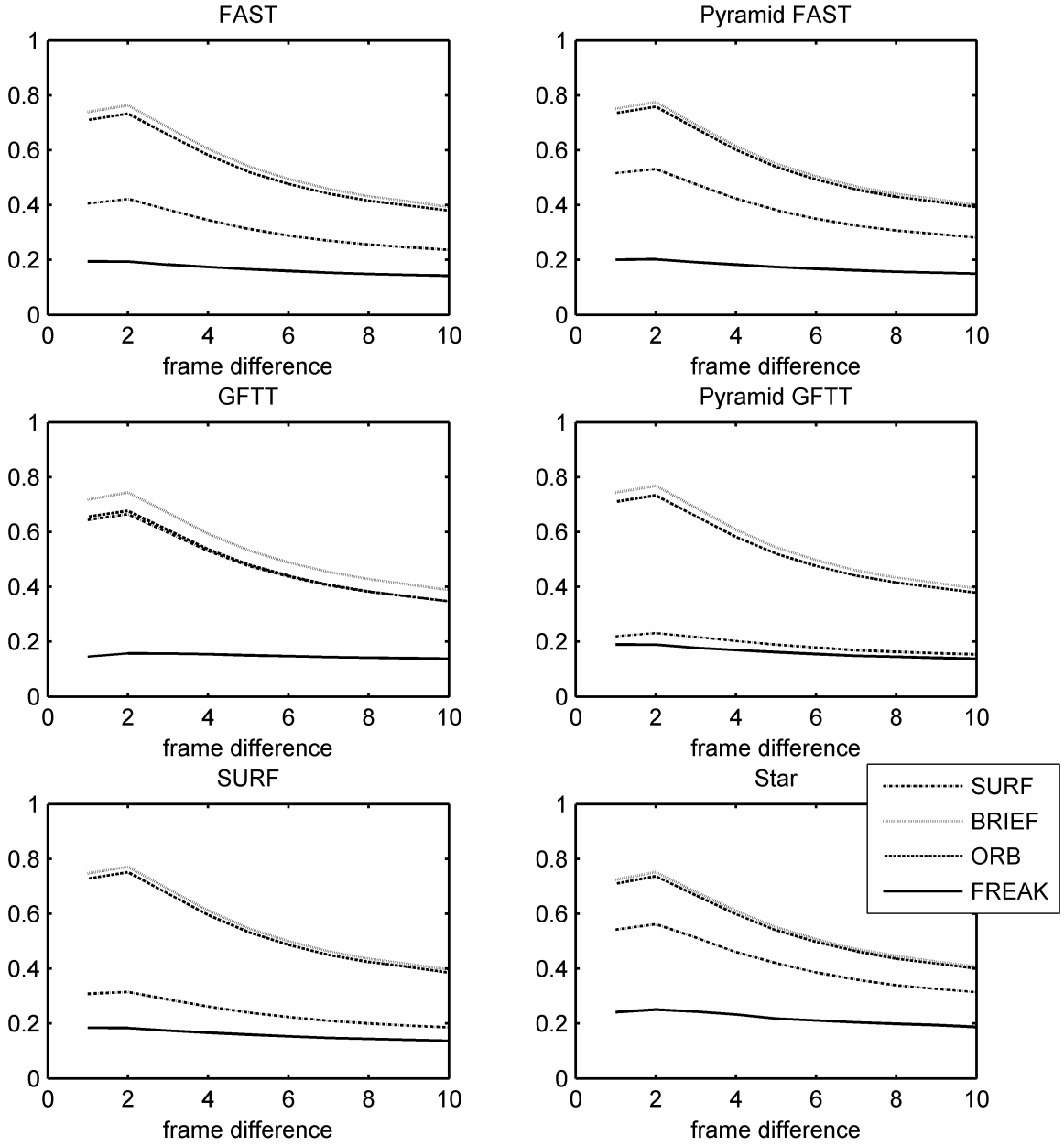

Fig. 15. Matching ratio of points detected with the analyzed detectors. Freiburg Dataset.

Conference of the Spanish Association for Artificial Intelligence, 2007.

[15] M. Ballesta, A. Gil, Ó. Martínez, O. Reinoso, "Local Descriptors for Visual SLAM". In: *Proc. Workshop on Robotics and Mathematics*, 2007.

[16] A. Schmidt, M. Kraft, A. Kasiński, "An evaluation of image feature detectors and descriptors for robot navigation", *Lecture Notes in Computer Science*, vol. 6375, 2010, pp. 251–259.

[17] H. Aanas, A. L. Dahl, K. S. Pedersen, "Interesting Interest Points – A Comparative Study of Interest Point Performance on a Unique data set", *International Journal of Computer Vision*, vol. 97, 2011, pp. 18–35.

[18] A. L. Dahl, H. Aanas, K. S. Pedersen, "Finding the Best Feature Detector-Descriptor Combination". In: *Proc. of 3DIMPVT*, 2011.

[19] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", *International Journal of Computer Vision*, vol. 60(2), 2004, pp. 91–110.

[20] J. Sturm, N. Engelhard, F. Endres, W. Burgard, D. Cremer, "Towards a benchmark for RGB-D SLAM evaluation". In: *Proc. of the RGB-D Workshop on Advanced Reasoning with Depth Cameras at Robotics: Science and Systems Conf. (RSS)*, 2011.

[21] J. Shi, C. Tomasi, "Good Features to Track". In: *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 1994, pp. 593–600.