An Evaluation of Image Feature Detectors and Descriptors for Robot Navigation

Adam Schmidt[1], Marek Kraft[1], and Andrzej Kasiński[1]

Poznań University of Technology, Institute of Control and Information Engineering,
Piotrowo 3A, 60-965 Poznań, Poland,
Adam.Schmidt@put.poznan.pl, Marek.Kraft@put.poznan.pl

Abstract. The detection and matching of feature points is crucial in many computer vision systems. Successful establishing of points correspondences between concurrent frames is important in such tasks as visual odometry, structure from motion or simultaneous localization and mapping. This paper compares of the performance of the well established, single scale detectors and descriptors and the increasingly popular, multiscale approaches.

1 Introduction

Many of today's computer vision applications – from mobile computers to robotic systems – require robust, markerless object detection. An increasingly popular approach to achieve this goal is the detection and matching or recognition of distinctive, natural image features across scene views. The developments in this field resulted in development of feature detectors and descriptors that can successfully cope with realistic application scenarios. In this paper, we investigate the usefulness of various detector-descriptor pairs in mobile robotics application. The motovation behind this is the fact, that in our opinion, characteristics of the features detected with a specific detector can significantly influence the matching process.

Specifically, we are interested in application of the feature detection and matching in mobile robot navigation (visual odometry [1] and SLAM [2]) in indoor environments. In this specific scenario, the inter-frame transformations of feature neighborhood used for description are relatively small, as the robot movement is mostly planar, and the enviroments is (in most cases) relatively feature-rich. The results we obtained can help to decide if the navigation process can (in the aforementioned circumstances) benefit from using more complex multiscale detectors and descriptors. The multi-scale approach, despite recent progress, is still time-consuming and the additional computational overhead may be unjustified in time-critical applications if it can be avoided.

2 Feature detectors and descriptors

This section contains short description of the investigated image feature detectors and descriptors.

2.1 Feature detectors

Harris corner detector. The popular Harris corner detector is based on investigating local auto-correlation function of the signal [3]. Variations of the auto-correlation function over principal directions are investigated. This is done using the so called structural tensor:

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

where I_x and I_y denote the partial image derivatives in x and y directions, respectively, and $w(p,q)$ denotes a weighting window over the area (p,q). Most common choice for the weighting window is a Gaussian, as it makes the responses isotropic. The criteria for cornerness measure is:

$$M_c = det(A) - \kappa trace^2 A \tag{2}$$

where κ is a tunable sensitivity parameter. Corners are the local maxima of M_c. Additional thresholding of the cornerness measure is applied to filter out the weak responses.

Shi Tomasi feature detector. The Shi Tomasi corner detector is strongly based on the Harris corner detector [4]. The difference lies in the cornerness measure M_c, which is computed as the minimum of the absolute values of the eigenvalues of the structural tensor (1).

$$M_c = min(|\lambda_1|, |\lambda_2|) \tag{3}$$

Corners are the local maxima of M_c. Such cornerness measure is said to be more robust than (2) while the image patches that surround features undergo affine transformations.

FAST corner detector. FAST stands for Features from Accelerated Segment Test [5]. The corner detector performs two tests. At first, candidate points are being detected by applying a *segment test* to every image pixel. Let I_p denote the brightness of the investigated pixel p. The test is passed, if n configuous pixels on a Bresenham circle with the radius r around the pixel p are darker than $I_p - t$ ('dark' pixels), or brighter than $I_p + t$ ('bright' pixels), where t is a threshold value. The authors use a circle with $r = 3$, and $n = 9$ for best results [5]. The ordering of questions used to classify a pixel is learned by using the ID3 algorithm, which speeds this step up significantly. As the first test produces many adjacent responses around the interest point, additional criterium is applied to perform a *non-maximum suppresion*. This allows for precise feature localization. The cornerness measure used at this step is:

$$M_c = max(\sum_{x \in S_{bright}} |I_{p \to x} - I_p| - t, \sum_{x \in S_{dark}} |I_p - I_{p \to x}| - t) \tag{4}$$

where $I_{p \to x}$ denotes the pixels laying on the Bresenham circle. As the second test is performed to only a fraction of image points that passed the first test, the processing time remains short.

SURF feature detector. SURF (Speeded Up Robust Features) is a image feature detector and descriptor [6], inspired by the SIFT detector [7], known for its high performance. SURF is designed with emphasis on speed, being SIFT's main weakness. SURF is said to be few time faster than SIFT with no performance drop. To achieve this, the detector uses the Haar wavelet approximation of the blob detector based on Hessian determinant. Haar wavelet approximations can be efficiently computed at different scales using integral images [10]. Accurate localization of features requires interpolation.

Star keypoint detector. Star keypoint detector is a part of OpenCV computer vision library. It was derived from CenSurE (Center Surround Extrema) feature detector introduced in [8]. The main motivation behind development of this detector was to achieve a full spatial resolution in a multiscale detector. According to [8], as SIFT and SURF detectors perform subsampling, the accuracy of feature localization is affected. CenSurE detector uses a bi-level approximation of the the LoG filter. The circular shape of the mask is approximated by shapes, which allow to combine simplicity and rotation invariance. CenSurE uses octagon shaped filter masks, while the star keypoint detector mask shape is made of two squares, one of which is rotated by 45 degrees (see figure 1). Such filters, regard-

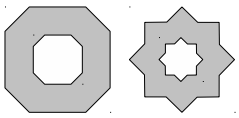

Fig. 1. Filters used by CenSurE (left) and star (right) keypoint detectors

less of size, can be efficiently computed using integral images. Instead getting scale-space by subsampling, the scale-space is constructed by applying filters of different size, thus allowing for precise localization without interpolation.

2.2 Feature descriptors

Correlation based similarity measures. Matching by correlation based similarity is achieved by taking a square window of certain size around the feature found in the reference image to find the best matching feature in another image – one with the highest or the lowest (depending on the measure used) similarity score. For our experiments we have chosen to use three common similarity measures: SAD (sum of absolute differences), SSD (sum of squared differences) and NCC (normalized cross-correlation) [9]. SAD and SSD are the simplest similarity measures, SSD being the more discriminative (image patch differences are amplified by squaring). NCC is more complicated, but is supposed to offer more robustness to lighting and contrast changes [9]. This is achieved by normalization.

SURF family of descriptors. The SURF family of descriptors encodes the distribution of pixel intensities in the neighborhood of the detected feature at the corresponding scale. The use of Haar wavelets in conjunction with integral images allows to decrease computation time [10]. The wavelets allow to determine the gradient values in x and y direction. Computation of the descriptor for a given feature can be divided into two distinct tasks.

The first task is the dominant orientation assignment. This steps allows to achieve the rotation invariance. First, the Haar wavelet responses are computed for all points within the radius $6s$ of the detected interest point, where s is the coefficient denoting the scale, at which the interest point was detected. The size of the wavelets is also properly rescaled – the side length is set to $4s$. The responses are then weighted with a Gaussian with the $\sigma = 2s$. Each response is a point in the vector space with the x-responses along the abscissa and the y-responses along the ordinate. The domimant orientation is selected by rotating a circle segment covering an angle of $\frac{\pi}{3}$rad around the origin. The responses in each segment are then summed and form a resultant vector. The orientation of the longest resultant vector is selected as the dominant orientation for the descriptor.

The construction of the descriptor itself starts with selecting a square window with the side size of $20s$ that is centered on the interest point and properly oriented (as computed in the previous step). The window is divided regulary into 4×4 square subregions. Each one of these subregions is divided into 5×5 regulary spaced sample points. Next, Haar wavelet response for each of the principal directions (dx, dy) is computed at each of these sample points. The absolute values of these responses are also incorporated into the descriptor. The directions are considered with respect to the dominant orientation. The complete form of the descriptor for each of the subregions is computed by summing the responses and their absolute values acquired at each of the sample points (see equation 5).

$$DESC_{sub} = [\sum dx, \sum dy, \sum |dx|, \sum |dy|] \tag{5}$$

The responses are weighted with a gaussian centered at the interest point to increase the robustness. Each subregion contributes with 4 elements of the descriptor vector. For 16 subregions, the descriptor size is therefore 64. Hence, the descriptor name is SURF64. The authors proposed also a simplified version of the descriptor, in which the window is divided into 3×3 square subregions. This reduced the descriptor's dimensionality and allowed for faster matching. The name of this variant is SURF36. If the rotational invariance is not required, the upright version of the descriptor, called USURF can be used. Skipping the dominant orientation assignment step yields further reduction of the time required for computation.

In our experiments, we used an additional modification to the original SURF descriptor proposed in [8]. To eliminate the boundary effects, the subregions are extended and overlap. The overlap size is $2s$. This is called MSURF, and its upright version – MUSURF.

3 Experiment description

The detector-descriptor pairs were evaluated on a sequence registered while driving a mobile robot around a lab room. The path was approximately 20 meters long. The sequence is one minute long and consists of 758 frames. The UI-1225LE-C uEye camera (IDS Imaging) with 1/3" CMOS sensor was used. The camera enables fast registration (up to 87 FPS) with the global shutter. The camera was equipped with a wide-angle lens ($f = 2.8mm$). No filtering and rectification of images was performed. To maintain stable framerate, exposure time was set to a constant value, but the automatic gain control function was used. The sequence displays many realistic effects – noise, distortions, contrast variations etc.

Each one of the described feature detectors was tested with each one of the described corner descriptors. No restrictions were made to the search area and no cross checking was performed – we always kept the feature in Image 2 that best matched the feature in Image 1. Matching was performed with five different threshold settings, to find the best compromise between the number of matched features and the match quality.

To evaluate the quality of matches, we used the 8-point algorithm for fundamental matrix computation [11], with robust estimation method based on the RANSAC (random sample consenus) algorithm [12]. RANSAC is an iterative method allowing for the estimation of a mathematical model (the fundamental matrix in our case) from a set of observations (matched points) containing outliers (false matches). The ratio of inliers (i.e. feature matches consistent with the estimated fundamental matrix) to all the detected matches is the indicator of the quality of the matching process. Additionally, any case in which the number of inliers was less than 10 was treated as a matching failure. The symmetric reprojection error [11] was used to check consistency of the data with mathematical model. To further examine the accuracy of the interest point location in images, the estimation using RANSAC was performed with two different thresholds: $\theta = 0.5$ pixel and $\theta = 1.0$ pixel. Example images from the sequence are given in figure 2.

Fig. 2. Example frames from the registered sequence

4 Results and discussion

Tables 1 and 2 show the results of running different detector-descriptor pairs on the test sequence with the threshold set to $\theta = 0.5$ and $\theta = 1.0$ pixels. The results are the ratios of inliers to all the detected matches averaged over the whole run. The detection and matching was performed for each consecutive frame pair. It's worth noting, that there were no failed matches (i.e. frames with fewer than 10 inliers) in this case. The results show, that the MSURF descriptor family out-

Table 1. Ratio of inliers to all detected matches, $\theta = 0.5$, frame to frame matching

	SAD	SSD	NCC	MSURF36	MSURF64	MUSURF36	MUSURF64
FAST	0.60	0.67	0.61	0.84	0.88	0.83	0.86
Harris	0.61	0.66	0.64	0.85	0.87	0.84	0.85
Shi-Tomasi	0.71	0.79	0.68	0.77	0.80	0.90	0.90
SURF	0.54	0.56	0.63	0.78	0.80	0.79	0.81
Star	0.61	0.65	0.71	0.74	0.75	0.73	0.73

Table 2. Ratio of inliers to all detected matches, $\theta = 1.0$, frame to frame matching

	SAD	SSD	NCC	MSURF36	MSURF64	MUSURF36	MUSURF64
FAST	0.77	0.83	0.76	0.93	0.95	0.935	0.95
Harris	0.79	0.83	0.80	0.95	0.96	0.953	0.96
Shi-Tomasi	0.88	0.92	0.83	0.91	0.94	0.973	0.98
SURF	0.64	0.65	0.72	0.90	0.92	0.912	0.92
Star	0.76	0.79	0.86	0.92	0.92	0.914	0.92

performs other descriptors. Another thing worth noting is that the differences in performance between MSURF descriptors with different dimensionality are relatively small. This leads to the conclusion, that for dense sampling it is safe to use the shorter descriptor. The multiscale detectors behave in general worse than Harris, FAST and Shi-Tomasi detectors while paired with the same descriptor. A few interesting phenomena were observed. The Shi-Tomasi feature detector, contrary to all the other detectors, performed better with the upright version of SURF family descriptors. The reason is probably the fact, that the detector responds also to edge-like features. Assigning dominant orientation is therefore harder than it is the case with the other detectors, responding to corner-like features. False orientation assignment leads to mismatches. As there are no significant in-plane rotations, the standard and upright versions of MSURF perform similarly in terms of ratios of inliers to all the detected matches averaged over the whole sequence. However, its versions that detect dominant rotation tend to produce more matches. Interestingly, the Star detector based on CenSurE, which we expected to perform better than SURF in terms of acccuracy (according to the claims in [8]) outperformed SURF only when paired with SAD, SSD and NCC matching schemes. However, it was outperformed by SURF when any

of the MSURF descriptors was used for matching with $\theta = 0.5$, which is the case that requires more accurate feature localization. Despite the changes in illumination and contrast, NCC is not apparently better than SAD or SSD. The exception to this rule is the Star detector, which seems to work well combined with NCC. Running the RANSAC estimation with different thresholds for allowed reprojection error ($\theta = 0.5$ and $\theta = 1.0$) does not prove that any of the described detectors is more accurate than any other, as no general conclusions holding for all matching schemes can be drawn. The Star detector combined with MSURF descriptors os the only exception, as it was mentioned earlier.
In our next experiment, the detection and matching was performed between every 10th frame of the sequence. As the SURF descriptor family achieved the highest performance in the previous test, SAD, SSD and NCC matching was not performed in this case. The Shi-Tomasi feature detector was not used, because it was impossible to achieve accuracy (in terms of average inlier number to match number ratio) on par with the other detectors without introducing matching failures (less than 10 inliers among the matches). The evaluation criteria remained the same. The results are shown in tables 3 and 4. The percentage

Table 3. Ratio of inliers to all detected matches, $\theta = 0.5$, every 10th frame matching

	MSURF36	MSURF64	MUSURF36	MUSURF64
FAST	0.51	0.57	0.53	0.57
Harris	0.52	0.58	0.54	0.56
SURF	0.47	0.52	0.49	0.52
Star	0.51	0.56	0.50	0.53

Table 4. Ratio of inliers to all detected matches, $\theta = 1.0$, every 10th frame matching

	MSURF36	MSURF64	MUSURF36	MUSURF64
FAST	0.67	0.73	0.72	0.75
Harris	0.70	0.75	0.73	0.75
SURF	0.65	0.69	0.66	0.69
Star	0.69	0.73	0.68	0.71

of inliers is clearly lower when compared with tables 1 and 2. Relaxation of matching threshold from 0.1 to 0.2 was necessary, to avoid frames with too few inliers. The performance difference between multiscale and single scale detectors is not as apparent as it is in the case of small inter-frame differences, but the single scale detectors still demonstrate better performance. An exception to this rule is the FAST detector combined with MSURF36. The the descriptor length seems to be more important as the baseline gets longer – 64-element descriptors offer better inlier ratio in this case. Interestingly, upright versions of the MSURF descriptors seem to perform better in terms of inlier ratio. However, the versions with dominant orientation detection return more matches. The results show, that also in the case of more significant displacements it is still safe to use single-scale detectors paired with the performant MSURF family descriptors. Multiscale algorithms would probably get better notes under longer

displacements and severe view angle change, scaling or rotations. Such extreme conditions are however uncommon in most mobile robotics applications.

5 Conclusions

We have presented the results of performance evaluation of different feature detector-descriptor pairs. The performance was tested with focus on indoor mobile robot navigation applications. The results show, that for this application domain, using single-scale feature detectors paired with the MSURF family descriptors does not introduce any performance penalty when compared to state of the art multiscale detectors. The use of single-scale approach allows to reduce the computational overhead. Futher reduction in processing time us possible by using the upright family of MSURF descriptors, as indoor applications usually do not introduce significant in-plane rotations in successive frames. Matching can be sped up by using the reduced version of the description vector. The effect on performance depends in this case on the inter-frame differences. All the tests were performed on unrectified, unfiltered images taken by a moving camera with a wide angle lens, without restrictions on matching area. Using additional criteria e.g. considering the ratio of distance from the closest match to the distance of the second closest [7] and reducing the search radius would improve the results.

References

1. David Nistèr, Oleg Naroditsky, James Bergen: Visual odometry. Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, 1, 652–659 (2004)
2. A.J. Davison, I.D. Reid, N.D. Molton, O. Stasse: MonoSLAM: Real-Time Single Camera SLAM. IEEE Transactions on PAMI, 29, 6, 1052–1067 (2007)
3. C. Harris and M. Stephens: A combined corner and edge detector. Proceedings of the 4th Alvey Vision Conference, 147–151 (1988)
4. J. Shi and C. Tomasi: Good Features to Track. Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, 593–600 (1994)
5. E. Rosten and T. Drummond: Machine learning for high-speed corner detection. Proc. of European Conf. on Computer Vision, 430–443 (2006)
6. H. Bay, A. Ess, T. Tuytelaars, L. Van Gool: SURF: Speeded Up Robust Features. Computer Vision and Image Understanding 110, 3, 346–359 (2008)
7. Lowe, D. G.: Distinctive Image Features from Scale-Invariant Keypoints International Journal of Computer Vision, 60, 2, 91-110 (2004)
8. M. Agrawal, K. Konolige, M.R. Blas: CenSurE: Center surround extremas for realtime feature detection and matching Proc. ECCV, LNCS 5305, 102–115 (2008)
9. J.Banks and P. Corke Quantitative Evaluation of Matching Methods and Validity Measures for Stereo Vision The Int. J. of Robotics Research, 20, 7, 512–532 (2001)
10. R. Lienhart and J. Maydt An Extended Set of Haar-like Features for Rapid Object Detection Proc. of Int. Conf. on Image Processing, 1, 900–903 (2002)
11. R. Hartley and A. Zisserman Multiple View Geometry in Computer Vision, 2nd ed. Cambridge University Press, 24, 381–395 (2004)
12. M.A. Fischler and R.C. Bolles Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Communications of the ACM, 24, 381–395 (1981)