Robustness of Point Feature Detection

Zijiang Song and Reinhard Klette

The *.enpeda..* Project, Department of Computer Science
The University of Auckland, New Zealand

Abstract. This paper evaluates 2D feature detection methods with respect to invariance and efficiency properties. The studied feature detection methods are as follows: Speeded Up Robust Features, Scale Invariant Feature Transform, Binary Robust Invariant Scalable Keypoints, Oriented Binary Robust Independent Elementary Features, Features from Accelerated Segment Test, Maximally Stable Extremal Regions, Binary Robust Independent Elementary Features, and Fast Retina Keypoint. A long video sequence of traffic scenes is used for testing these feature detection methods. A brute-force matcher and Random Sample Consensus are used in order to analyse how robust these feature detection methods are with respect to scale, rotation, blurring, or brightness changes. After identifying matches in subsequent frames, RANSAC is used for removing inconsistent matches; remaining matches are taken as correct matches. This is the essence of our proposed evaluation technique. All the experiments use a proposed repeatability measure, defined as the ratio of the numbers of correct matches, and of all keypoints.

1 Introduction

A diversity of 2D or 3D feature or keypoint detection methods has been proposed in computer vision in recent years. For a 2008 review on 2D feature detectors, see [15], and for current evaluations of 3D keypoint detectors, see [14, 16].

The different methods have their own advantages and disadvantages. This paper focuses on 2D feature detection methods which are implemented in OpenCV (version 2.4.4) [11]. We test the robustness of these feature detection methods with respect to rotation, Illumination changes, scaling, and blur.

Our motivation for this test arose when we had to decide for 2D or 3D feature detectors for accurate ego-motion analysis in a driver-assistance context, and for unifications of partially 3D-reconstructed surfaces (during different runs) obtained from recorded stereo street views. Because of the existence of the above cited 3D evaluations, we only focus on the 2D case in this paper.

Figure 1 illustrates video data as recorded in a driver-assistance context. Due to the limited length of this paper, we only use the illustrated video sequence (of 400 stereo frames, recorded at 25 Hz) for the reported tests in this paper. Due to actually occurring changes in recorded videos (with respect to brightness or blurring), and due to changes in pose of recorded objects (mainly with respect to scale, but also with respect to rotation), we are interested in 2D keypoints which

Fig. 1. Sample of an image of the discussed video sequence (*top*, the black pixels at the border are due to rectification of the stereo frames), with four images after processing, illustrating blurring (*middle, left*), rotation (*middle, right*), brightness changes (*bottom, left*), and scaling (*bottom, right*). The used sequence of 400 stereo frames is in Set 4 of EISATS [5], and called "Cyclist"; 640×480 images are recorded with 10 bit per pixel.

can be robustly tracked in the presence of variations in brightness, blurring, scaling, or rotation. Basically, the provided features for those keypoints should allow to do such a robust tracking.

The rest of the paper is structured as follows: Section 2 gives a brief introduction for the used feature detectors. Section 3 discusses the design of our experiments. Section 4 informs about our experimental results. Section 5 concludes the paper.

2 Used Feature Detectors

We briefly introduce the used keypoint detectors, together with features for those keypoints.

SIFT. The *scale-invariant feature detector* (SIFT) was published in 1999; see [7]. It consists of four major stages: scale-space extrema detection, keypoint localization, orientation assignment, and keypoint description. The first stage uses difference-of-Gaussians (DoG) to identify potential interest points, which were invariant to scale and orientation. DoG is used instead of the Laplacian to improve computation speed. In the keypoint localization step, the operator rejects low contrast points and eliminates edge response. The Hessian matrix is used to compute the principal curvatures and to eliminate keypoints that have a ratio between both principal curvatures that is greater than a threshold.

An orientation histogram is formed from gradient orientations of sample points within a region around the keypoint (defined by the scale of the keypoint) in order to get an orientation assignment. It was suggested that best results are achieved with an 4×4 array of histograms, with eight orientation bins in each. Thus, the SIFT descriptor is a vector of $4 \cdot 4 \cdot 8 = 128$ dimensions.

MSER. The detection of *maximally stable extremal regions* (MSER) was published in 2002; see [10]. It is used as a method of blob detection in images, for example to find correspondences between image elements from two images with different viewpoints. A new set of image elements, that are put into correspondence, are called *extremal regions* that have two important properties. The set is closed under (1) continuous transformations of image coordinates (i.e. affine transformations, warping, or skewing), and (2) monotonic transformations of image intensities. However, the approach is known to be sensitive to natural lighting effects such as change of day light, or moving shadows.

SURF. The detector of *speeded up robust features* (SURF) was presented in 2006; see [2]. SIFT and SURF algorithms employ slightly different ways for detecting features. SIFT builds an image pyramid, filters each layer with Gaussians of increasing sigma values, and takes the differences. SURF is inspired by the SIFT detector, but designed with emphasis on speed, being SIFT's main weakness. SURF is often said to be "a few times faster than SIFT with no performance drop". The detector uses a Haar-wavelet approximation of the blob detector based on the Hessian determinant. Haar-wavelet approximations can be efficiently computed at different scales using integral images. Due to the use of integral images, SURF filters the stack using a box-filter approximation of second-order Gaussian partial derivatives, since integral images allow the computation of rectangular box filters in near-constant time. Accurate localization of features requires interpolation.

FAST. The detection of *features from accelerated segment test* FAST was also published in 2006; see [12]. It 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 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 $r = 9$ for best results. 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, an additional criterion is applied to perform a non-maximum suppression. This allows for precise feature localization. The cornerness measure used at this step is as follows:

$$M_c = max(\sum_{x \in S_{bright}} |I_{p \to x} - I_p| - t, \sum_{x \in S_{dark}} |I_p - I_{p \to x}| - t) \qquad (1)$$

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

BRIEF. Binary robust independent elementary features (BRIEF) have been suggested in 2010; see [3]. This is is a general-purpose feature point descriptor that can be combined with arbitrary detectors. It uses binary strings for efficiency reasons. The descriptor is highly discriminative even when using relatively few bits and can be computed using simple intensity difference tests. It is robust to typical classes of photometric and geometric image transformations. Similarity between descriptions can be evaluated using the Hamming distance, which is very efficient to compute, instead of using the usual L_2-norm. BRIEF is targeting real-time applications leaving them with a large portion of available CPU power for subsequent tasks but also allows running feature point matching algorithms on computationally weak devices such as mobile phones.

BRISK. The detector of *binary robust invariant scalable keypoints* (BRISK) is a method for keypoint detection, description and matching, published in 2011; see [6]. In this paper, a comprehensive evaluation on benchmark datasets reveals BRISK's adaptive, high-quality performance compared to state-of-the-art algorithms, albeit at a dramatically lower computational cost (an order of magnitude faster than SURF in many cases). The key to speed lies in the application of a novel scale-space FAST-based detector in combination with the assembly of a bit-string descriptor from intensity comparisons retrieved by dedicated sampling of each keypoint neighbourhood.

ORB. The detection of *oriented binary robust independent elementary features* (ORB) was also published in 2011; see [13]. It is a standard for oriented FAST and rotated BRIEF. The algorithm uses FAST in pyramids to detect stable keypoints, selects the strongest features using FAST or a Harris response, finds their orientation using first-order moments, and computes the descriptors using BRIEF (where the coordinates of random-point pairs (or k-tuples) are rotated according to the measured direction).

FREAK. The *fast retina keypoint* (FREAK) detection was published in 2012; see [1]. The algorithm proposes a novel keypoint descriptor inspired by the human visual system, and, more precisely, the retina. A cascade of binary strings is computed by efficiently comparing image intensities over a retinal sampling pattern. It is commonly stated that FREAKs are in general faster to compute with lower memory load and also "more robust" than SIFT, SURF, or BRISK, and that they are competitive alternatives to existing keypoints, in particular for embedded applications.

In the brief descriptions above we also mentioned "common believe" about the performance of those detectors, and we are now aiming at quantifying such statements by experiments on extensive data. For providing repeatable data, we use a data set available online on [5].

3 Experiment Design

We are interested in comparative evaluations of efficiency (the time used for a measurement of feature points and description extraction time), rotation invariance (how the feature detection method depends on feature direction), scaling

invariance (how the feature detection method depends on feature size), blur invariance (how the feature detection method is robust against blur), and illumination invariance (how the feature detection method is robust against illumination changes).

All the reported quality tests work in a similar way: For a given sequence (our set of source images), we apply the defined workflow identically on each image, and take finally the average of calculated data for the whole image sequence. Here we report for experiments on the sequence "Cyclist", see Fig 1, which is a "typical" day-time sequence with respect to occurring diversities in shown objects and lighting variations.

For any image in the sequence, the following steps are done for each of the tested eight feature detectors:

1. Read the source image I_s as a greyscale image.
2. Use the feature detector to detect the keypoints and extract descriptors D_s from I_s; get the number K_s of keypoints.
3. Transform the source image for the different invariance test scenarios:
   - For testing rotation invariance, rotate I_s around its centre in steps of 1 degree; we obtain 360 transformed images.
   - For testing scaling invariance, resize I_s in scaling steps of 0.01, from $0.25\times$ size to $2 \times size$, thus calculating 175 scaled images.
   - For testing illumination invariance, we change the overall image brightness by adding a scalar to every pixel value of I_s; the scalar changes from -127 to 127 in stepsize 1, thus generating 255 transformed images.
   - For testing blurring invariance, we Gaussian blur I_s by using 20 different kernel sizes between 3 to 41, thus calculating 20 blurred versions of I_s.
4. For each of those transformed images I_t, we use the feature detector again to detect the keypoints and extract descriptors D_t; in particular we note the number K_t of keypoints.
5. We use the two sets of descriptors D_s and D_t and a brute-force descriptor matcher to find matching image keypoints between source image and transformed image.
6. There are inconsistent matches. We use *Random Sample Consensus* (RANSAC) to remove those; all the remaining are considered to be *correct matches*. We note the number K_{cm} of correct matches.

The *repeatability measure* $r(I_s, I_t)$ is defined as being the ratio between the number of correct matches between the two sets of image keypoints (for source and transformed image) and the number of keypoints detected in the source image:

$$r(I_s, I_t) = \frac{K_{cm}}{K_s}$$

Obtained measure values are then averaged for all the selected frames of the used test sequence. Here we report about results for 90 randomly selected images from this sequence. For the feature detectors we used the default parameters implemented in OpenCV.

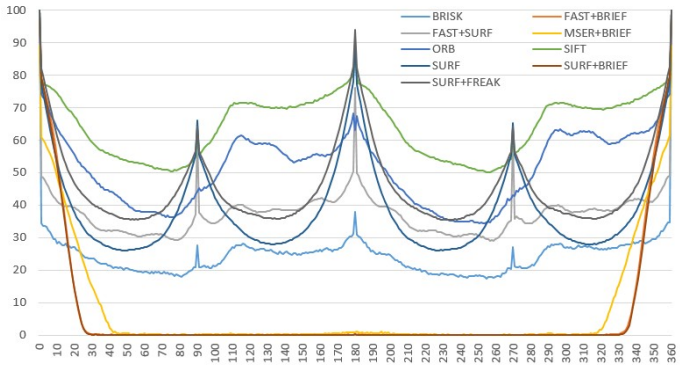

Fig. 2. Repeatability values for the tested 360 rotations.

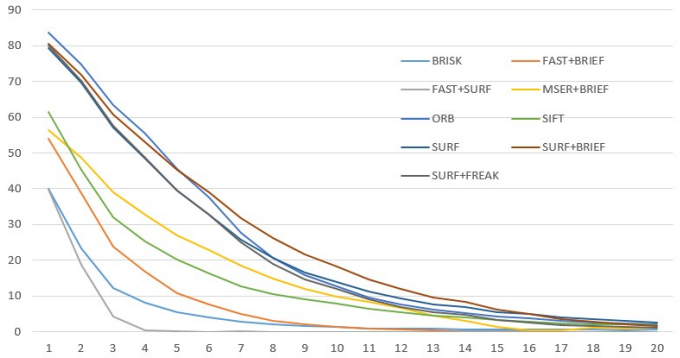

Fig. 3. Repeatability values for the tested 20 blurrings.

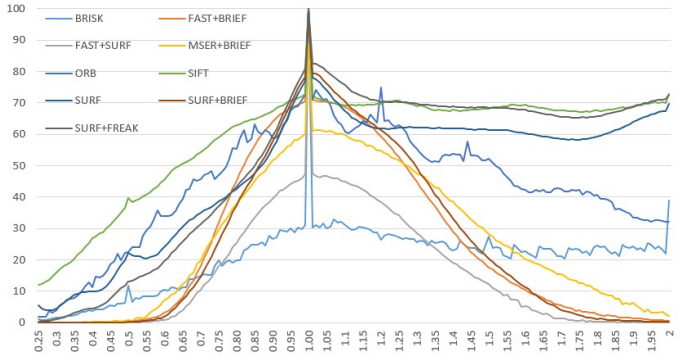

Fig. 4. Repeatability values for the tested 175 scalings.

4 Experimental Results

The following four graphs in Figs. 2 to 5 summarize our experimental results. The y-axis is always for values $r_{(I_s,I_t)}$. The x-axis for the blur graph is a number which indicates the kernel size ($2\times$ number $+1$) of the used Gauss function.

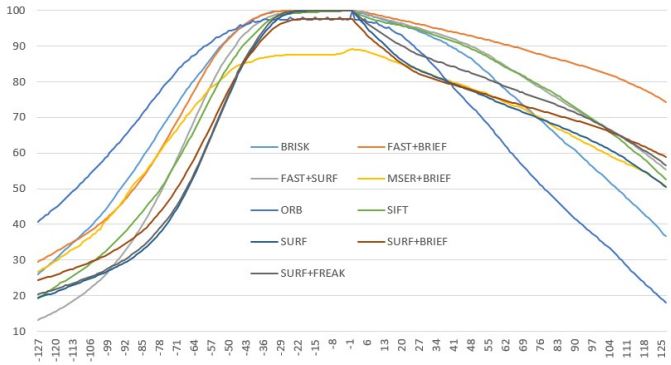

Fig. 5. Repeatability values for the tested 255 brightness variations.

Method	Ave. time per frame	Ave. time per keypoint	No. of keypoints
SIFT	254.1	0.55	726
SURF	401.3	0.40	1,313
ORB	9.6	0.02	500
BRISK	8.5	0.06	258
SURF+BRIEF	101.1	0.10	1,259
MSER+BRIEF	55.1	0.52	116
FAST+BRIEF	4.3	0.01	1,451
FAST+SURF	51.9	0.05	1,590
SURF+FREAK	96.3	0.19	847

Table 1. Averages are for the detectors on all the 90 source images, and the numbers of keypoints are for the image shown in Fig. 1.

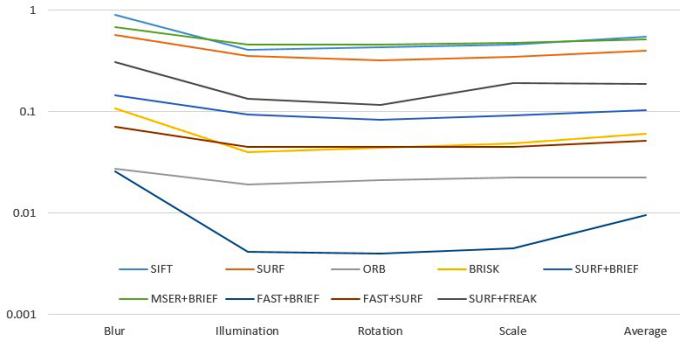

Fig. 6. Averaged time per keypoint for the four generated sets of transformed images, averaged over all the selected 90 frames. The up-axis shows the time used in milliseconds.

When naming a feature detection method 'A+B' then this means that we use feature detector 'A' and feature descriptor 'B'. For the brightness variations note that the brightness mean for all the used 90 source images equals 108.

Table 1 summarizes time measurements on original images, and illustrates numbers of detected keypoints for the image shown in Fig. 1. The measured computation times per keypoint in all the generated test images are summarized in Fig. 6.

5 Conclusions

This paper compared the performance of eight feature detection methods. The performed tests show that SIFT has the best robustness with respect to rotation and scale changes, but its time issue has been confirmed again. FAST and BRIEF provide better results for increased brightness, and ORB better for decrease brightness. ORB also shows good performance on blurred images. Many more comments are possible for the given graphs, but the reader may see for himself. The results also show that claimed invariances are only valid to some limited degree, and further research on improving invariance properties appears to be justified.

References

1. Alahi, A., Ortiz, R., Vandergheynst, P.: FREAK: Fast retina keypoint. In Proc. CVPR (2012) 510–517
2. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: Speeded up robust features. In Proc. ECCV (2006) 404–417
3. Calonder, M., Lepetit, V., Strecha, C., Fua, P.: BRIEF: Binary robust independent elementary features. In Proc. ECCV (2010) 778–792
4. Donoser, M., Bischof, H.: Efficient maximally stable extremal region (MSER) tracking. In Proc. CVPR (2006) 553–560
5. EISATS Website : http://www.mi.auckland.ac.nz/index.php. Last visited in April (2013)
6. Leutenegger, S., Chli, M., Siegwart, R.Y.: BRISK: Binary robust invariant scalable keypoints. IEEE Int. Conf. ICCV (2011) 2548–2555
7. Lowe,D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Computer Vision **60** (2004) 91–110
8. Luo J., Oubong G.: A comparison of SIFT, PCA-SIFT and SURF. Int. J. Image Processing (2009) 143–152
9. Martin A. F., Robert C. B.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Comm. ACM **24** (1981) 381–395
10. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. Proc. BMVC (2002) 384–396
11. OpenCV Documentation: www.docs.opencv.org/index.html. Last visited in April (2013)
12. Rosten, E., Drummond, T.: Machine learning for high-speed corner detection. In Proc. ECCV (2006) 430–443
13. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: ORB: An efficient alternative to SIFT or SURF. In Proc. ICCV (2011) 2564–2571
14. Tombari, F., Salti, S., Di Stefano, L.: Performance evaluation of 3D keypoint detectors. Int. J. Computer Vision **102** (2013) 198–220
15. Tuytelaars, T., Mikolajczyk, K.: Local invariant feature detectors: A survey. Foundations Trends Computer Graphics Vision **3** (2008) 177–280
16. Yu, T.-H., Woodford, O.J., Cipolla, R.: A performance evaluation of volumetric interest point detectors. Int. J. Computer Vision **102** (2013) 180–197