Machine Vision and Applications (2015) 26:443–466
DOI 10.1007/s00138-015-0679-9

ORIGINAL PAPER

A comparative experimental study of image feature detectors and descriptors

Dibyendu Mukherjee[1] · **Q. M. Jonathan Wu**[1] · **Guanghui Wang**[2]

Received: 19 September 2013 / Revised: 2 November 2014 / Accepted: 2 March 2015 / Published online: 18 April 2015

Abstract Feature detection and matching is a fundamental problem in many computer vision applications. In the past decades, various types of feature detectors and descriptors have been proposed in the literature. Although several comparative studies on feature detectors and descriptors have been performed in the past, few studies have been carried out concerning recently proposed descriptors such as BRISK, FREAK, etc. Also, previous comparisons were either application oriented or limited in experimentation or in the number of detectors and descriptors compared. This paper provides a comprehensive review of a large number of popular feature detectors developed in the last three decades. The study makes several contributions to the development of a generic comparison of feature detectors and descriptors. First, we conduct comparisons of invariance against image transformations such as illumination changes, blurring, rotation, scaling, viewpoint changes, exposure, JPEG compression, combined scaling and rotation, and combined viewpoint changes. Second, we provide a proper distinction between detectors and descriptors using separate comparisons. Third, a few detectors have been tested on the variation of parameter values. Fourth, we conduct a statistical analysis of invariance against four popular types of transformations: viewpoint changes, blurring, scaling, and rotation. Fifth, we carry out intuitive matching between detectors and descriptors, testing on simulated and practical scenarios. Last, we conduct exhaustive experiments on several datasets for each combination of detectors and descriptors to provide a ranking that can also be weighted to suit specific applications.

Keywords Local features · Feature detectors · Feature descriptors · Comparative study

✉ Dibyendu Mukherjee
mukherjd@uwindsor.ca

Q. M. Jonathan Wu
jwu@uwindsor.ca

Guanghui Wang
ghwang@ku.edu

[1] Department of Electrical and Computer Engineering, University of Windsor, 401 Sunset Avenue, Windsor, ON N9B 3P4, Canada

[2] Department of Electrical Engineering and Computer Science, University of Kansas, Lawrence, KS 66045, USA

1 Introduction

A feature in an image refers to specific meaningful structure in the image. Features can range from a single pixel to edges and contours, and can be as large as objects in the image. Feature detection is the process of detecting these meaningful structures in an image. The output of a feature detector is usually a number of specific locations in an image, called feature points. These locations are chosen based on their tolerance against noise, transformations, and other deformations. A feature detector often has a descriptor for its feature points. A feature descriptor represents either a subset of the total pixels in the neighborhood of the detected feature points or other measures generated from the feature points. In literature, some well-known feature detectors provide feature descriptors. Thus, in the rest of the paper the detectors that are comprised of descriptors will be mentioned in the respective sections.

Since the very beginning of computer vision, feature detectors have occupied an important place in the research due to their numerous applications in such areas as object recognition, categorization, classification, robot localization and tracking, image matching and 3D reconstruction, image retrieval, registration, etc.

Due to the increasing demand for better feature detection in various fields of application, a question of suitability arises. Not all types of feature detectors are suitable for a specific application. Thus, application-specific comparative studies of feature detectors demand more attention. Although, there have been many papers on this subject, only a few of them compare feature detectors against common types of image transformations and deformations, and very few of them include most of the state-of-the-art feature detectors. Also, a separate comparison for descriptors has not been carried out in general.

This work aims to provide a comparison guide for some of the recently proposed feature detectors and descriptors. First, nine most popular feature detectors and descriptors are briefly reviewed. Then, each detector–descriptor pair is evaluated under various transformations. A number of popular datasets consisting of natural transformations are used for the evaluation, and simulated transformations are used where databases provide insufficient data. Furthermore, we attempt to intuitively match a detector with a descriptor and compare their performance on real and simulated data. Finally, we compare these matched detectors and descriptors on motion tracking using a large number of real-life road image sequences. The findings of this study, as discussed at the end of Sect. 2, make several new contributions to this field. We outline these contributions after a brief discussion of the previous works to establish a proper relation.

The organization of this paper is as follows: Sect. 2 provides an overview of the background and establishes the necessity for our current research. Section 3 discusses the feature detectors and descriptors. Section 4 provides the experimentation details. Finally, the paper is concluded in Sect. 5.

2 Background

The research on feature detectors and descriptors is a fast growing area in image processing. The following short review has been arranged chronologically to explain the gradual improvements in feature detection, as well as the evolution and limitations of comparative studies.

The first corner detection algorithm was designed by Moravec [34]. Harris and Stephens [19] revealed the limitations of this detector with their popular corner and edge detector (henceforth named Harris corner detector as it is popularly called). Lucas and Kanade proposed a popular method, further developed by Tomasi and Kanade [44], often called the Kanade–Lucas–Tomasi feature tracker or KLT tracker, based on Harris and Stephens detector. A comprehensive review of a number of popular detectors is presented in [39]. The authors compare Harris corner detector to a number of other algorithms [9,21]. Later, Schmid et al. [40] revised the comparison method originally proposed, and conducted a number of qualitative tests. The test results indicate that the Harris operator (which is the basis of Harris corner detector) is the best among the compared methods. Shi and Tomasi [41] proposed a new detection metric (Good Features To Track or GFTT) based on the Harris operator, arguing that their model was a better choice. A completely new approach, called SUSAN, was also proposed in [42]. However, the proposed operators of SUSAN fall short when rotation and scaling are involved. Hall et al. [18] provided a definition of saliency under change of scale and evaluated Harris, the method proposed in [28] and Harris–Laplacian corner detector [31]. Harris–Laplacian is a combination of the Harris and Laplacian function for characteristic scale selection. Inspired by the need for a scale-invariant approach, Lowe [29] proposed one of the most popular feature detectors: the Scale Invariant Feature Transform or SIFT. SIFT is a combination of feature point detector and descriptor. Mikolajczyk and Schmid [32] compared the SIFT descriptor and a number of related descriptors, proving SIFT to be one of the best feature detectors based on the strength of its descriptor. Also, Zuliani et al. [48] provided a mathematical comparison of Harris, KLT, and a number of popular approaches. Harris and SIFT have been well explored and improved by several researchers [7,10,27].

Moreels and Perona [35] compared Harris, Hessian, and difference of Gaussian filters on images of 3D objects with different viewpoints, lighting variations, and scale variations. They differentiated between detectors and descriptors, using SIFT, PCA-SIFT, steerable filters, and shape context descriptors. They found a good match of detector and descriptor. In recent years, a number of reviews have been conducted. Tuytelaars and Mikolajczyk [45] provide an informative review of many popular feature detectors, and good comparisons are provided in [16,26]. However, new comparison reviews are required to include newly developed detectors and descriptors.

In 2012, Ziegler et al. [47] proposed a simple descriptor, comparing it to BRIEF and SURF, both of which are discussed in the following section. Also, Heinly et al. [20] and Khvedchenia [23] reviewed some of the popular feature detectors and descriptors. However, the shortcomings of the existing research lie in the limitations of experiments, the lack of proper distinction, and comparison of so few feature detectors, descriptors and their combinations. Studies similar to our proposed one are carried out by Aanæs et al. [1], Dahl et al. [6], and Kaneva et al. [22]. Studies in [1] and [6] are well explained but are limited in terms of the number of compared detectors and descriptors. Kaneva et al. [22] provide a study mostly concentrated on synthetic transformations. Also, there are a number of evaluation studies [8,15] of feature detectors and descriptors for specific applications. However, a study covering both general evaluations and application-specific evaluations remains unavailable.

In the current literature, several excellent feature detectors and descriptors have been proposed. The unified study by Moreels and Perona [35] to find an optimal detector–descriptor pair is noteworthy. However, as the number of detectors and descriptors to compare increases, such a study becomes time-consuming and cumbersome. Due to the incorporation of various detectors and descriptors that perform differently for different applications, such a study often depends on applications and implementation approach. However, as the search for appropriate and accurate feature detectors and descriptors is a continuous process, updated comparative studies are always required. Our effort is directed at updating the comparative studies in conjunction with existing literature. In the process, we focus on experimental evaluations and analyses that are less highlighted in individual studies. The following statements present a summary of the relation of the proposed work to earlier studies, and the features that distinguish our study from related works.

1. A number of recent feature detectors and descriptors are included in addition to those already compared in [1,6,20,23]. This increases the number of comparisons to a very large extent, since our effort is to compare each detector–descriptor combination.
2. In addition to using the databases used by Heinly et al. [20], we have added two new databases with more datasets to make a fair and reliable comparison. Finally, we combine each detector with each descriptor to provide a rich comparison.
3. We include both experimental results and a statistical analysis in this paper. For our experiments, a number of popular metrics are used in line with [20]. We have also introduced a new metric, Reliability, to evaluate performances using a new practical framework, discussed next.
4. We provide a practical framework on motion extraction to test the performance of a number of intuitively chosen detector–descriptor pairs. Our main goal behind this is to demonstrate an evaluation of a number of detectors and descriptors in practice. Unlike previous studies, we provide both an experimental validation and a practical validation. This framework can provide numerous practical scenarios and serve as a guideline and a stepping-stone for future research.
5. We have designed a testing framework to judge the statistical significance of our comparisons in accordance with the work of Dahl et al. [6]. A *t* test based on the Precision metric has been performed on four major transformations, namely viewpoint changes, blurring, scaling, and rotation. Compared to [6], we have included more detectors and descriptors and covered multiple transformations on a large number of images. In addition, we have reported the hypothesis, variance, and degrees of freedom to benefit the study of future researchers.
6. We provide a comparison and discussion on performance variation due to parameter changes for a number of detectors. Although only a few of the detectors are studied due to limited space, this idea can be extended to most of the detectors and descriptors.
7. Finally, we propose a ranking methodology for the detector–descriptor combinations in terms of individual transformations, individual metrics, and overall combinations. In comparison to the rankings used in previous studies, the proposed ranking is more generic and can be used to include or exclude any transformation and/or metric. Also, a threshold-based positional ranking is introduced to maintain a similar ranking for detector–descriptor combinations with comparable performances. Due to its generic nature, this ranking can be used in application-specific environments and in generalized settings.

The study presents a comparative experimental study of different feature detectors and descriptors. It can benefit researchers in choosing a better detector and descriptor combination for feature matching and act as a connection between previous studies and future work in feature detection and matching.

3 Brief overview of feature detectors

The following subsections discuss the feature detectors and descriptors compared in this paper. We have endeavored to cover different types of detectors and descriptors in terms of methodology, historical significance, popularity, and age. The detectors and descriptors are ordered according to the time when they were proposed. SIFT is the oldest and FREAK is the most recent in the literature.

3.1 SIFT

Proposed by Lowe [29], SIFT is one of the most popular feature detectors and descriptors in the literature. SIFT can efficiently identify object points in noisy, cluttered, and occluded environments due to its high invariance to translation, scaling, and rotation. In this method, interest points are extracted from the image in two steps. First, the image is repeatedly smoothed using Gaussian filters and subsampled to find images in smaller scales. This way, an image pyramid is constructed with the reference image at the ground level (level 1). Second, interest points are discovered in the $3 \times 3 \times 3$ neighborhood of any pixel at an intermediate level. These points are obtained from the image points where the difference-of-Gaussians values attain an extrema, both in spatial domain and at the scale level of the Gaussian pyramid.

The interest points extracted this way show scale invariance and rotation invariance.

Lowe has also proposed a descriptor for these interest points. The descriptor is a position-dependent histogram of local image gradient directions around the interest point, and is also scale invariant. There are numerous extensions of the basic SIFT method, such as PCA-SIFT, color-SIFT, etc. The SURF method is also largely based on SIFT, as discussed in Sect. 3.4.

3.2 MSER

In computer vision, Maximally Stable Extremal Regions (MSER), proposed by Matas et al. [30], are used as a method to detect blobs in images. A fast implementation of MSER can be found in [36]. Extremal regions have two properties: (a) the set of these regions is closed under affine or projective transformations on an image; (b) the set is closed under transformations like lighting variations. Thus, they are scale and rotation invariant as well. These regions are detected using a connectivity analysis and by computing connected maximal and minimal intensity regions.

3.3 FAST

Rosten and Drummond [37] propose Features from Accelerated Segment Test (FAST). Corners are detected in a number of training images, first using FAST and then through machine learning, to determine the best criteria for detection. We find that the criteria for detection are decisions about whether or not the pixel is a corner. The criteria together create a decision tree, which can correctly classify all the corners in the training images, embedding the rules for detecting a corner from any test image. The decision tree is then converted into a C-code, which is used as a corner detector.

3.4 SURF

Speeded Up Robust Features (SURF) is a very efficient and robust scale and rotation-invariant feature detection and descriptor algorithm proposed by Bay et al. [4]. It is very similar to SIFT and is based on a Hessian matrix, which is generated by convolution of the Gaussian second-order derivative with image pixels. The interest points are extracted in the same way as SIFT. This is accomplished by a $3 \times 3 \times 3$ non-maximal suppression on a Gaussian pyramid, followed by interpolation of the maxima of the Hessian matrix.

The SURF detector is found on each interest point by orientation assignment and descriptor component analysis. The orientation is assigned by calculating a Haar Wavelet response in x and y directions in a circular neighborhood of each interest point. The dominant orientation is found by calculating the sum of orientations. Then, the Wavelet responses in a square region oriented in the dominant orientation provides the SURF descriptors. These descriptors are scale and rotation invariant and are very robust against transformations on images.

3.5 CENSURE

Center Surround Extremas (CENSURE) is a very accurate detector proposed by Agrawal et al. [2] based on two criteria: stability (persistence of features across viewpoint changes) and accuracy (consistency of feature localization across viewpoint changes). The method first determines the maxima using a method called Hessian–Laplacian. The Laplacian is approximated using center surround filters called bi-level filters. Using this Laplacian, a basic center-surround Haar Wavelet is formed. These are the CENSURE responses. A non-maximal suppression provides the necessary features.

3.6 BRIEF

Binary Robust Independent Elementary Features (BRIEF), developed by Calonder et al. [5], is also a recently developed attractive descriptor of binary strings that is very useful for extracting descriptors from feature points for image matching. BRIEF is also used in our experiments as a common descriptor for those detectors that do not have their own descriptors: FAST, CENSURE, and MSER. BRIEF has fast execution and good accuracy.

3.7 BRISK

Binary Robust Invariant Scalable Key-points (BRISK), proposed by Leutenegger et al. [25], is a detector and a descriptor. In this method, points of interest are first identified in the image pyramid using a saliency criterion. Next, a sampling pattern is applied to the neighborhood of each of these detected key points to retrieve gray values, which are then used to generate the orientation. The oriented BRISK sampling patterns provide the descriptor. Once generated, these key points can be matched very efficiently due to the binary nature of the descriptor.

3.8 ORB

The Oriented FAST and Rotated BRIEF (ORB) detector and descriptor was proposed by Rublee et al. [38] as a very fast alternative to SURF. The authors state that ORB provides the following [38]: (a) addition of a fast and accurate orientation component to FAST; (b) efficient computation of oriented BRIEF features; (c) analysis of variance and correlation of oriented BRIEF features; and (d) a learning method for de-correlating BRIEF features under rotational invariance, leading to a better performance.

3.9 FREAK

Fast Retina Key-point (FREAK) is a very recently developed descriptor, proposed by Alahi et al. [3] and inspired by the human visual system (HVS), or more precisely, by the retina. First, a retinal sampling pattern is generated using a circular sampling grid called the "retinal sampling grid", which has a higher density of points near the center. Next, a binary descriptor is formed by a sequence of one-bit difference of Gaussians (DoG). A human vision-like search called a "saccadic search" is used to select relevant features. Last, the rotation of the key points at each selected feature point is computed using the sum of the local gradients over selected pairs, similar to BRISK.

4 Experiments

This section includes comparisons of the detectors and descriptors based on their performances in image matching under specific distortions. Two images are matched for their similar regions. The images compared are referred to as the "Reference" image and the "Target" image. The target image has a level of distortion or transformation in comparison to the reference, leading to a possible mismatch. Our goal is to determine the accuracy of matching with varying degrees of distortion. The experiments section has been divided into a number of subsections based on the analysis, the target of the experiments, and the practicality of the experiments.

1. **Datasets**: we have made every effort to keep our work compatible with existing studies, while including more datasets with different transformations. We use the Oxford datasets provided by Mikolajczyk et al. [33], Herz-JesuP8 and Fountain-P11 datasets from Strecha et al. [43], and other datasets from Heinly et al. [20]. We also include the Amsterdam Library of Object Images (ALOI) [14] and an image database from the Signal and Image Processing Institute or USC-SIPI [46] to increase the number of test images with a goal of providing a consistent and robust review based on a large number of images. A subset of these images are shown in Fig. 1 grouped according to their distinctiveness. Finally, we also use a number of simulated transformations to demonstrate the performance on scaling, blurring, and rotation. The types of transformations in the datasets used are described below:
   - *Oxford datasets*: image blurring, exposure, JPEG compression, combined scaling, rotation, and perspective transformations of planar geometry;
   - *fountain-P11 and Herz-JesuP8 datasets*: perspective transformations of non-planar geometry;
   - *Datasets from Heinly et al.*: pure rotation, scaling, illumination changes;
   - *ALOI*: illumination variation and viewpoint changes;
   - *USC-SIPI*: rotation of textures.
2. **Performance metrics**: we use four metrics for performance evaluation, as described below.
   - *The Putative Match Ratio = number of putative matches/number of features*: is used to address the selectivity of the descriptor. This represents the fraction of detected features to be initially identified as a match. This ratio is based on the matching criteria. With over-restrictive matching, this ratio will decrease. Of course, feature point detection capability varies from one detector to another, as depicted in Fig. 2. Hence, the selectivity of the descriptor and consequently, this metric plays an important role in the evaluation of a detector-descriptor pair.
   - *Precision = number of correct matches /number of putative matches*: the number of correct matches is found by geometrically verifying the initial putative matches based on a known camera position. This is also driven by how restrictive the matching criterion is. However, this metric also represents the matching accuracy of a detector–descriptor pair. High precision indicates a better pair, as only correct matches would be used for feature matching. For a specific image pair, the precision may also be zero, indicating that no putative matches were correct. This fact is used in defining our last metric: Reliability.
   - *Matching Score = number of correct matches /number of features*: this is equivalent to the multiplication of Putative Match Ratio and Precision and indicates how the descriptor has performed in extracting the number of correct matches from the number of initially detected features. This can also be evaluated as a metric measuring the pairing strength between a detector and a descriptor. A descriptor has a higher affinity with a detector if the descriptor has a better selectivity over feature points detected, increasing the Matching Score.
   - *Reliability = number of test image pairs with non-zero precisions/total number of test image pairs*: this metric indicates the fraction of image pairs with at least one correct match out of the total number of image pairs used. This does not depend on the quality of matching, but is useful in practice where matches are required between every two pairs of images. By two pairs, we mean two reference and target image pairs. An example is shown in Sect. 4.7, where a pair of reference and target images are represented by a stereo pair (left and right, respectively) and two pairs are found by two subsequent frames (a frame contains

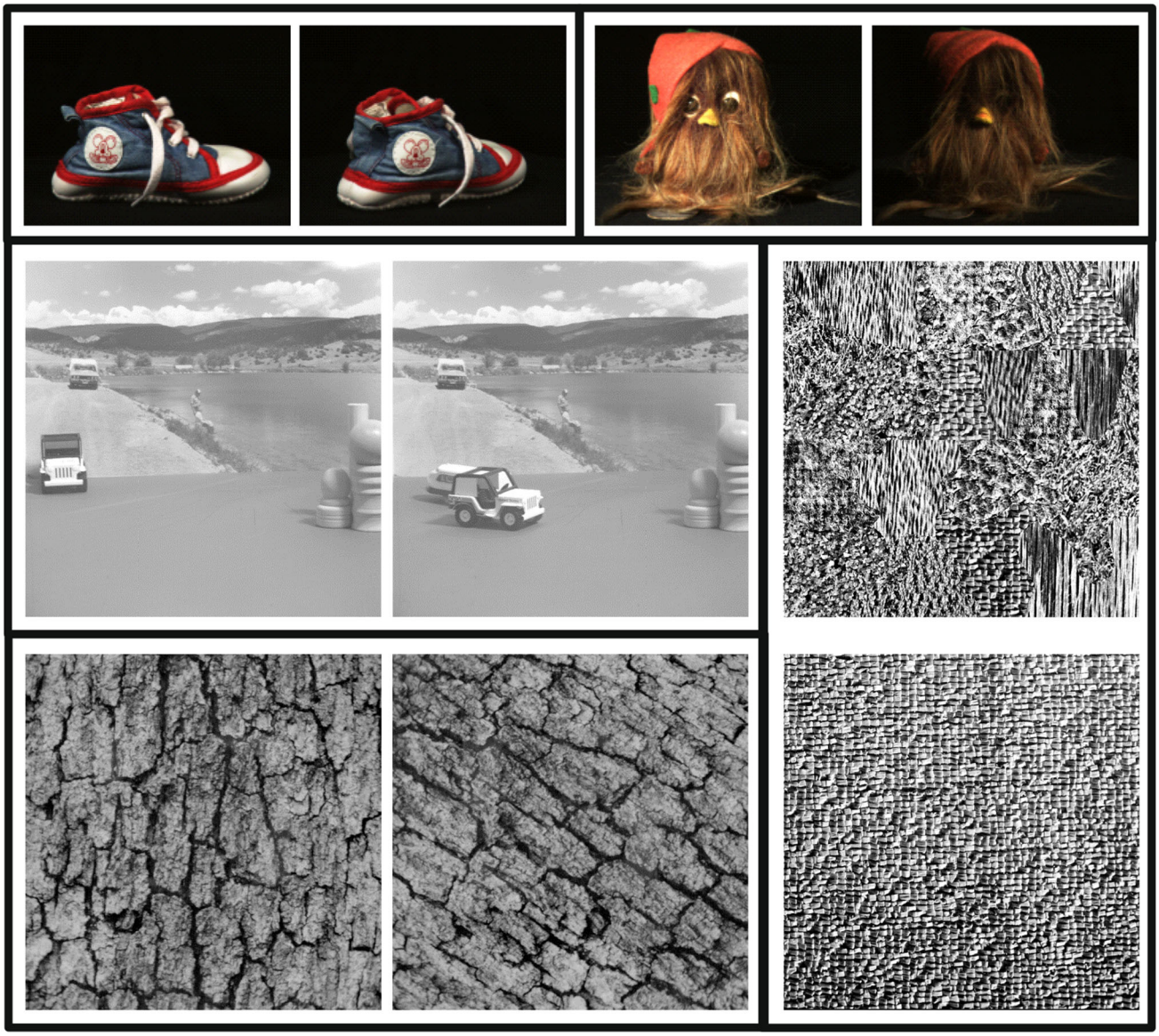

Fig. 1 A subset of the images used for the experiment. Images are grouped to *highlight* their distinctiveness. In clockwise direction from *top-left*, the first group shows two images with viewpoint changes, the second group shows illumination variation, the third group consists of some texture images from USC-SIPI used for simulated transformations, the fourth group shows rotational transformation, and the last group shows two images from a sequence

a stereo pair). Highly reliable detector–descriptor pairs should be able to find correct matches for a large number of images over several transformations. This is crucial in our practical scenario discussed in Sect. 4.7, where insufficient matching in a single pair may lead to significant localization errors. Also, based on the requirements of an application, this metric can be formed using a threshold on the precision value. For example, Reliability may be formed by dividing the number of test images with a precision above a certain threshold by the total number of test images. Of course, this would represent a stricter condition.

- *Mean Execution Time*: this is found by averaging the average time taken to process each feature point over all transformations. Mean execution time becomes important for a real-time application; thus, it is also used for rank computation.

3. **Test setup**: Table 1 classifies feature detection and description methods based on the presence of feature detectors and descriptors. As presented in the table, a few methods are either detector or descriptor while others are both. Thus, a fair comparison in terms of matching is difficult. The best way to compare is to pair up each detector with each descriptor and compare the performance in

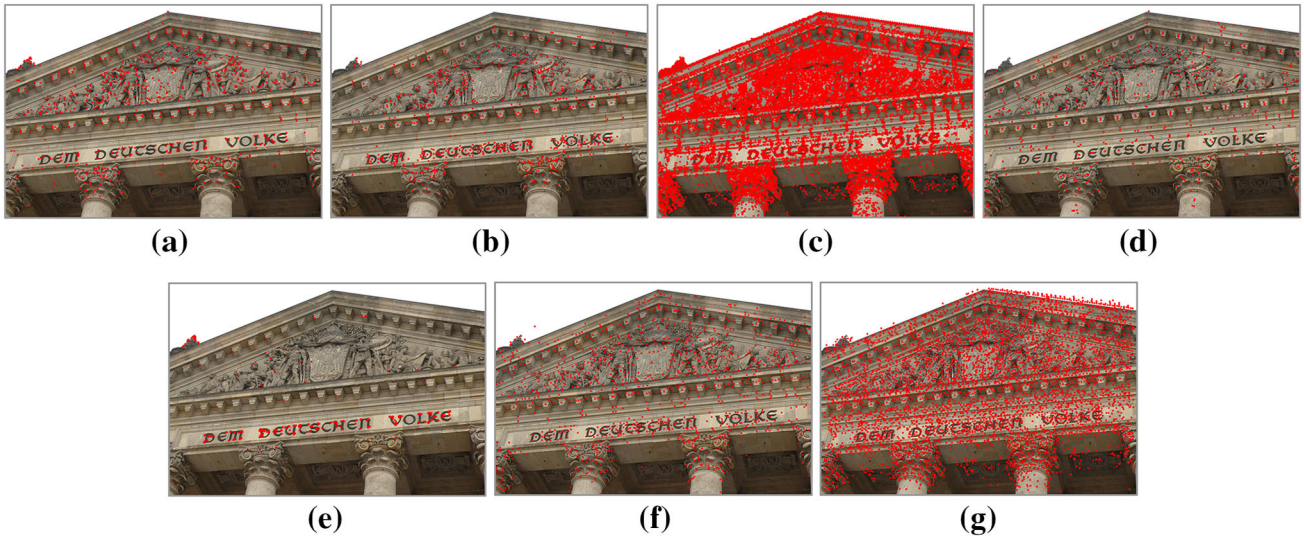

Fig. 2 Figure showing the appearance and distribution of key points from each detector: **a** BRISK, **b** CENSURE, **c** FAST, **d** MSER, **e** ORB, **f** SIFT, and **g** SURF. Evidently, FAST detects a lot of feature points, whereas ORB detects the lowest number of feature points. The other detectors have ranges in between these two detectors

Table 1 Comparison of methods in terms of availability of feature detector and descriptor

	BRIEF	BRISK	CENSURE	FAST	FREAK	MSER	ORB	SIFT	SURF
Detector	No	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes
Descriptor	Yes	Yes	No	No	Yes	No	Yes	Yes	Yes

terms of a number of metrics. We have evaluated each detector–descriptor pair on the referred datasets. For each dataset, the features are detected on reference and target images using the paired up detector. We then describe the features using the paired up descriptor. Finally, the features are matched using a brute-force matcher. As pointed out in [20], pairing up a detector with a descriptor introduces certain complications. The scale information makes a mismatch if either the detector or the descriptor is scale invariant while the other is not. If both the detector and descriptor are scale invariant, we discard the scale information and compute the descriptor in the original image resolution, where possible. A similar procedure is followed when both detector and descriptor provide orientation information.

4. **Match criteria**: The test uses brute-force matching to match the features in image pairs. No restrictions have been imposed on the number of features detected. The default parameter values defined in the OpenCV implementations of the algorithms have been used for the experiments. The matched feature points are reprojected onto the source image using the homography or the ground truth of 3D geometry (in case of non-planar geometry) provided, and a reprojection error is calculated. In cases where ground truth information is not provided, the transformation values are used to reproject the matched points onto the reference image. Transformation values are represented by the amount of transformation applied to the reference image to construct the target image, and these values are known specifically for simulated transformations. A reprojection error is the Euclidean distance between the original and the reprojected feature point. A threshold of 2 on the reprojection error has been used to limit the inliers, i.e., correct matches. The threshold is empirically chosen following [20]. However, it is reduced to provide a stricter condition. The threshold cannot be too low, as there will always be noisy measurements and reprojection errors. Finally, the matcher uses L2 norm (Euclidean distance) or the Hamming distance between the descriptors for matching. L2 norm is suitable for SIFT and SURF, while the Hamming distance is more suitable for ORB, BRISK, BRIEF, and FREAK. We have tested with both types of distances and followed the choice of distance as mentioned above, to maximize the performance of the descriptors. The performances of detectors and descriptors are reported in Figs. 3 and 4 and are discussed in detail in the following subsections.

For the rest of the document, we denote detector–descriptor pairs by a general notation: "Detector+Descriptor". If both the detector and descriptor are part of a single feature detection and/or description technique, they are simply represented by the name of the technique. For example: FAST+SIFT represents a combination of FAST detector and

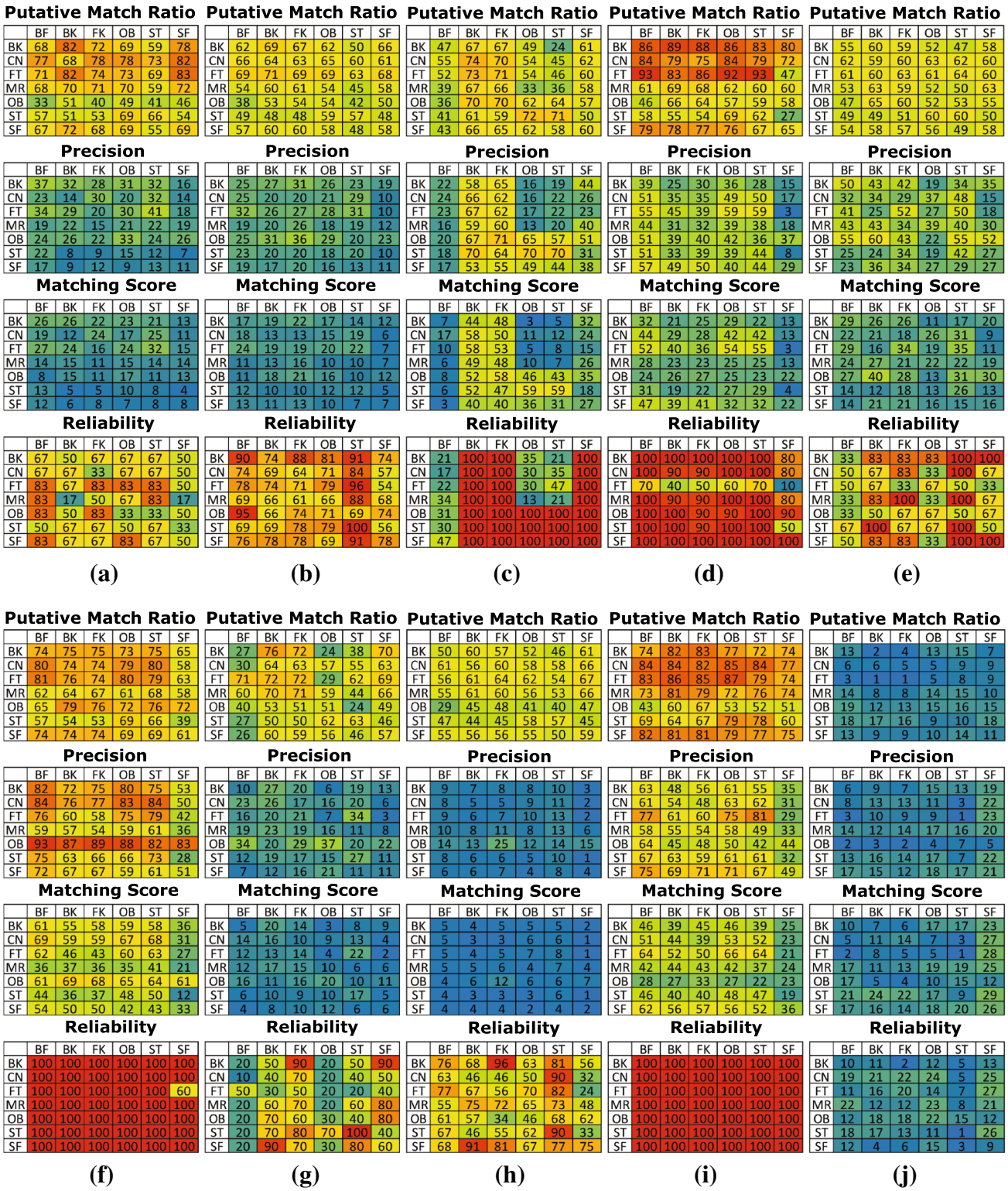

Fig. 3 Results for **a** illumination, **b** viewpoint changes, **c** rotation, **d** blur, **e** scale, **f** JPEG compression, **g** scale and rotation, **h** scale, rotation, and viewpoint changes, **i** exposure, **j** average ranking over all transformations. *Rows* are detectors, *columns* are descriptors: *BF* BRIEF, *BK* BRISK, *CN* CENSURE, *FT* FAST, *FK* FREAK, *MR* MSER, *OB* ORB, *ST* SIFT, *SF* SURF. Values are in percentages except for the ranks (*blue* 0 %, *red* 100 %) (color figure online)

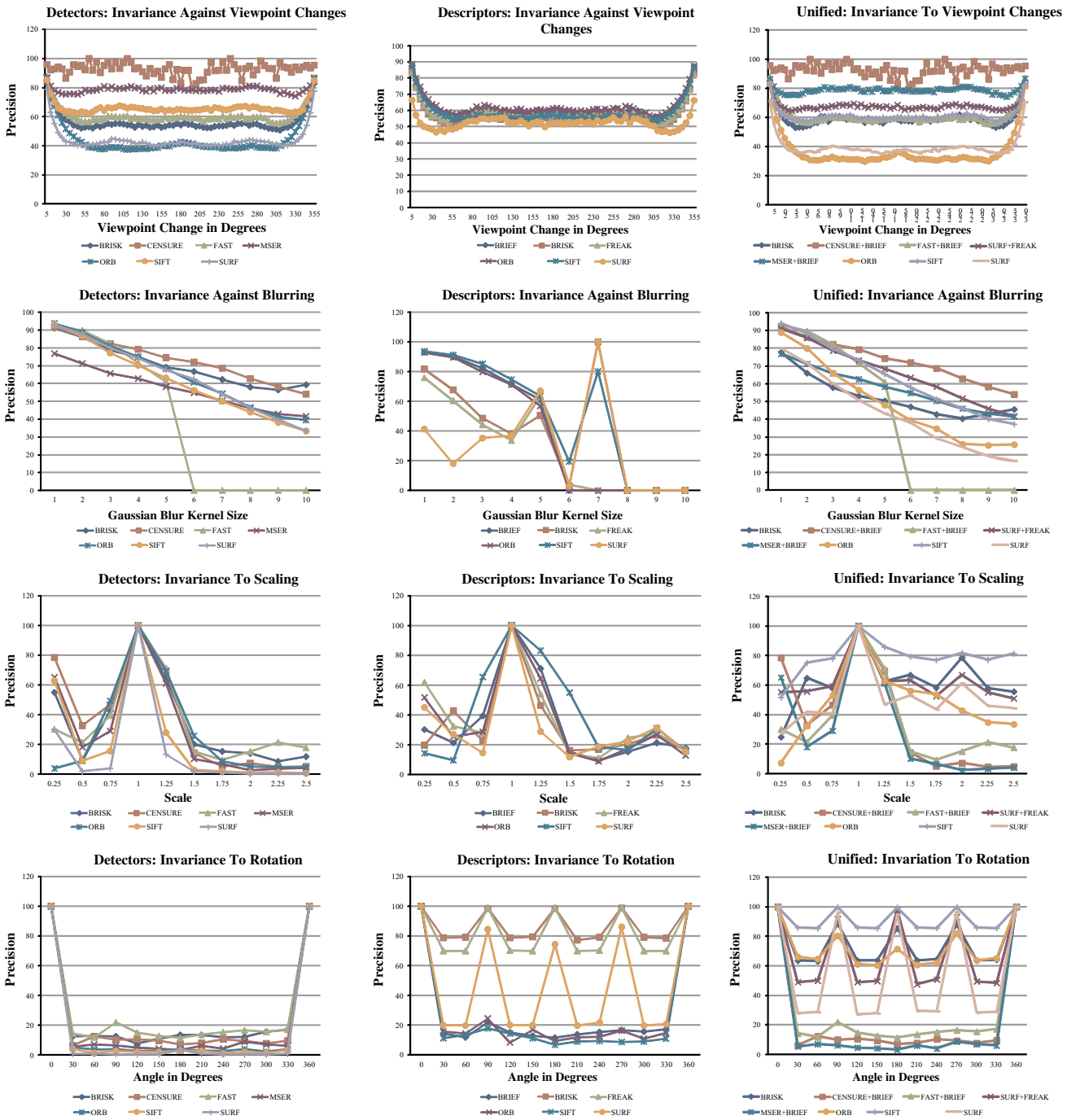

Fig. 4 Test of invariance against viewpoint changes (*first row*), blurring (*second row*), scaling (*third row*), and rotation (*last row*). The *first*, *second*, and *third columns* represent the results for detectors, descriptors, and unified combinations, respectively

SIFT descriptor, while SIFT represents a combination of SIFT detector and SIFT descriptor. The rest of Sect. 4 is divided as follows: the performances of different detector and descriptors are analyzed in Sects. 4.1 and 4.2, respectively. Figure 3 provides a consolidated average for each pair of detectors and descriptors over all transformation values and all datasets. Figure 4 shows the individual performance of detectors keeping a common descriptor, and performance of descriptors keeping a common detector, against gradually varying transformation values. Also, a few pairs are intuitively chosen for comparison based on the same framework, and the results are discussed in Sect. 4.3. The statistical significance of the results is analyzed using the Student's *t* test in Sect. 4.4. A study has been carried out on a few detectors for

a number of parameters, as presented in Sect. 4.5. This part of the experiment can also be extended to the rest of the detectors or descriptors. Next, based on the paired performances, a ranking methodology has been followed and individual ranks for each detector–descriptor pair are computed in Sect. 4.6. Finally, a practical scenario for benchmarking is described in Sect. 4.7.

4.1 Performance of detectors

Two figures are used to demonstrate the performances of detectors and descriptors. Figure 3 presents the cumulative performance of all combinations of detectors and descriptors on each type of transformation over all transformation values and datasets. This figure presents a holistic comparison. On the other hand, Fig. 4 shows the individual performances of each detector, keeping the descriptor the same, and the individual performances of each descriptor, keeping a common detector, with respect to the variations of transformation values. Finally, a number of detector–descriptor combinations are intuitively chosen to show more performance variations.

Before discussing detector performances for varying transformation values, we take a closer look at their cumulative performances by different detectors from Fig. 3. CENSURE performs very well in the presence of illumination, blur, jpeg, and exposure. As described in [2], the detector is built upon surrounding extremas, and it is proven to be more robust in comparison to SIFT or SURF. However, due to its high selectivity in finding robust feature points, its Reliability is quite low. On the other hand, ORB, SIFT, SURF, and FAST have shown moderate Precision, but yield higher Reliability. FAST produces a large number of feature points as shown in Fig. 2, hence it has high Reliability. However, feature points have low resistance against scaling, as can be seen in Fig. 3. Putative Match Ratios of FAST are high for most of the transformations, as it can find more feature points. However, after matching, FAST shows medium performance for scaling.

ORB produces fewer feature points compared to other detectors (as visualized in Fig. 2), and according to the table, they are robust against rotation and scaling. The experimental results support the claim in [38] that ORB is strong against rotation and scaling. BRISK has comparable performance and a high Reliability due to its internal use of FAST. SIFT and SURF do not perform well due to their partial incompatibility with the descriptors. They perform better in the presence of their own descriptors. However, SIFT performs better against rotation if paired with a rotationally invariant descriptor, specifically SIFT or ORB. ORB has a conflict with SIFT, and the orientation assignment by SIFT has not been used for ORB. Thus, this improved performance is due to ORB's resistance to rotation.

To find out more about the individual performances of the detectors, each detector has been used to perform against four fundamental transformations, namely viewpoint changes, blurring, scaling, and rotation. Their performances have been plotted in Fig. 4 (left column) against changing values of transformation levels using Precision as the measure of performance. Viewpoint transformations are provided in the ALOI database for 1000 images. Blurring, scaling, and rotation for increasing values of transformations are not part of any database; thus, simulated transformations have been used. The original images from ALOI viewpoint datasets have been used for the simulated transformations. For blurring, a Gaussian blur filter was applied to each image with kernel sizes in the range [1, 10] with steps of 1. For scaling, each image was resized using a scale factor in the range [0.25, 2.5] with steps of 0.25. Finally, for rotation, each image was rotated around its center, in a range of [0, 360]° with step size of 1°.

A common descriptor is used for all experiments. To prevent undue advantages, we cannot choose a descriptor already associated with a detector, leaving only BRIEF and FREAK. FREAK has shown very good performance with FAST and SURF. The descriptor uses the sum of local gradients, similar to BRISK, and it has a descriptive performance that is highly compatible with SURF (can be verified from results). Any SURF-like detector would also have high compatibility with FREAK. This compels us to use a descriptor like BRIEF that would not boost up the performance of some of the detectors, biasing the comparison. BRIEF also has no restriction from pairing with rotation invariant detectors. However, as the descriptor is mostly responsible for enhancing the feature detector by extracting rotation and illumination invariant descriptors, a descriptor such as BRIEF that is truly disassociated with any detector would be unable to enhance the capabilities of the detectors. This can be seen in the performance graphs in Fig. 4 (left column), where robust detectors like SIFT do not show expected performances.

4.2 Performance of descriptors

As shown in Fig. 3 for cumulative performance comparisons, BRISK and FREAK have shown very good performances in the presence of rotation, illumination changes, and viewpoint changes. They also have high Reliability. This signifies their robustness to these transformations, as described in the relevant papers [3,25]. FREAK also shows a surprisingly high compatibility with SURF. FAST and ORB also make a good combination, as ORB uses oriented FAST features. However, this combination does not perform well in the presence of rotation due to the loss of processing of FAST features for rotational invariance.

For similar reasons, FAST does not match well with SIFT. For SURF, as its methodology uses Hessian matrices and these matrices have good responses for corner features, FAST is more suitable. BRIEF is fast and has a moderate performance for most transformations. However, its lack of invariance to rotation is prominent.

As before, the performances of descriptors have also been compared against viewpoint changes, blurring, scaling, and rotation while keeping a common detector. The performances are shown in Fig. 4 (middle column). FAST has been used as an independent detector, mostly because it can produce a higher number of feature points for any type of transformation, as evident from the Putative Match Ratios present in Fig. 3. The results are more informative compared to the case of detectors. As evident from the figures, SIFT, SURF, and FREAK have high performance against scaling and blurring. ORB also has a considerably good descriptor.

4.3 An unified performance

Looking at the cumulative values from Fig. 3 and the Precision values over all transformations in Fig. 4, we find several good matches and bad combinations of feature detectors and descriptors. A variation of performance of all combinations or detector–descriptor pairs over all transformation values would be a valuable insight. However, due to limited space, we have intuitively chosen a few pairs among the long list of 42 pairs. Among the pairs, BRISK (detector and descriptor both belong to BRISK), ORB, SIFT, and SURF have been chosen, as they have performed well when paired with "themselves". We wanted to take at least one pair so that we could cover all the detectors and descriptors. With this in mind, CENSURE, FAST, and MSER have been used while pairing with BRIEF. BRIEF is used mostly because it is fast and has no orientation requirements; thus, it can be paired with any of the detectors. Also, as seen in the performances of the detectors with BRIEF, the performances of CENSURE, FAST, and MSER are good compared to SIFT or SURF paired with BRIEF. Finally, FREAK has shown high suitability when paired with SURF. In total, 8 pairs have been chosen for a unified performance: BRISK, CENSURE+BRIEF, FAST+BRIEF, SURF+FREAK, MSER+BRIEF, ORB, SIFT, and SURF.

We have chosen a few pairs out of the 42 pairs to study the invariance, as seen in Fig. 4, due to their use for practical purposes. To limit the number of experiments and the paper length, only a few pairs are chosen for comparison in the practical scenario described in Sect. 4.7. Of course, a detailed study on invariance can always be carried out on the same scenario for every pair. However, this small study serves as a highlight for such an extensive study. Before moving to Sect. 4.7, we have tested the chosen unified pairs on the same test bench as the detectors and descriptors. From Fig. 4 (last column), CENSURE+BRIEF, MSER+BRIEF, SURF+FREAK, and SIFT perform well against viewpoint changes. As scaling or rotation was not involved, the performance of BRIEF did not drop. However, for scaling and rotation, SIFT has shown the best performance. BRISK, SURF+FREAK, and ORB have also shown good performances. These results also support the results in Fig. 3, where these pairs performed well.

4.4 Statistical significance

A number of tests have been carried out on each detector–descriptor pair under varied situations. The metric values alone are insufficient to compare the performance of the pairs due to their large variations in different tests. Also, according to [6], the performance is affected by three factors, including problem difficulty, type of method used for experiment (the detector–descriptor pair in our case), and the noise introduced to computation. To determine whether the performance variations of each pair are due to noise or mutual differences, we carry out a paired Student's t test. The first factor in the test is to decide on the methods, i.e., the combination of detectors and descriptors to compare. Thus, in the rest of this section, a method denotes a "combination of a detector and a descriptor". As there are many combinations to compare, the test is carried out on detectors keeping the same descriptor (BRIEF, as before) and on the descriptors keeping same detector (FAST, as before). In this way, the combinations reduce the effective number of candidates for statistical comparison. The second factor in the test is the metric to consider for comparison. In [6], the authors created the receiver operating characteristic (ROC) curve and used the area under this curve (AUC) as the measure for the t test. We have used four metrics for the experiments and preferred to use precision as the metric, as it is among the most popular and common measures of performance. Finally, tests have been carried out on the results of viewpoint changes, scaling, blurring, and rotation. The procedure followed is provided next:

1. The mean value of precision μ over all parameter values of the experiment is considered for each method. For example, the mean Precision for 0–360 degrees rotation represents μ. The underlying assumption of the test is that the variance is the same for the two methods compared. Our goal is to understand whether the metric values found for two methods belong to the same distribution.
2. The null hypothesis: for two methods, 1 and 2, the means are identical, i.e., $\mu_1 = \mu_2$; thus, the methods are similar in performance.

Table 2 Statistical results for detectors on viewpoint (hypothesis index, followed by *t* value and SD)

	BRISK	CENSURE	FAST	MSER	ORB	SIFT	SURF
BRISK	0	1	1	1	1	1	1
	0	2.76E−52	2.69E−58	4.39E−46	5.61E−29	1.10E−37	1.34E−58
	0.00	7.12	0.79	5.45	5.31	3.24	1.87
CENSURE	1	0	1	1	1	1	1
	2.76E−52	0	6.97E−48	4.81E−38	1.96E−46	3.53E−51	1.27E−57
	7.12	0.00	7.15	4.47	11.52	5.38	7.88
FAST	1	1	0	1	1	1	1
	2.69E−58	6.97E−48	0	8.12E−39	2.91E−38	4.37E−20	1.83E−68
	0.79	7.15	0.00	5.49	5.34	3.27	1.91
MSER	1	1	1	0	1	1	1
	4.39E−46	4.81E−38	8.12E−39	0	2.88E−39	4.59E−50	9.81E−55
	5.45	4.47	5.49	0.00	10.55	2.66	6.19
ORB	1	1	1	1	0	1	0
	5.61E−29	1.96E−46	2.91E−38	2.88E−39	0	2.81E−33	8.42E−01
	5.31	11.52	5.34	10.55	0.00	8.31	5.19
SIFT	1	1	1	1	1	0	1
	1.10E−37	3.53E−51	4.37E−20	4.59E−50	2.81E−33	0	8.58E−56
	3.24	5.38	3.27	2.66	8.31	0.00	3.78
SURF	1	1	1	1	0	1	0
	1.34E−58	1.27E−57	1.83E−68	9.81E−55	8.42E−01	8.58E−56	0
	1.87	7.88	1.91	6.19	5.19	3.78	0.00

DOF: 70

Table 3 Statistical results for detectors on blurring (hypothesis index, followed by *t* value and SD)

	BRISK	CENSURE	FAST	MSER	ORB	SIFT	SURF
BRISK	0	1	1	1	0	1	0
	0	3.62E−02	1.46E−02	7.59E−09	6.63E−02	1.10E−02	5.87E−02
	0.00	3.52	32.01	2.04	8.11	8.96	9.31
CENSURE	1	0	1	1	1	1	1
	3.62E−02	0	9.86E−03	1.58E−10	7.84E−03	1.70E−03	6.12E−03
	3.52	0.00	32.26	1.60	7.53	8.44	8.09
FAST	1	1	0	0	1	1	1
	1.46E−02	9.86E−03	0	1.21E−01	1.21E−02	2.34E−02	1.57E−02
	32.01	32.26	0.00	31.99	25.40	24.90	25.70
MSER	1	1	0	0	1	0	1
	7.59E−09	1.58E−10	1.21E−01	0	9.75E−03	1.57E−01	3.14E−02
	2.04	1.60	31.99	0.00	7.58	8.50	8.46
ORB	0	1	1	1	0	1	0
	6.63E−02	7.84E−03	1.21E−02	9.75E−03	0	5.60E−05	1.48E−01
	8.11	7.53	25.40	7.58	0.00	1.63	2.02
SIFT	1	1	1	0	1	0	1
	1.10E−02	1.70E−03	2.34E−02	1.57E−01	5.60E−05	0	3.64E−03
	8.96	8.44	24.90	8.50	1.63	0.00	2.16
SURF	0	1	1	1	0	1	0
	5.87E−02	6.12E−03	1.57E−02	3.14E−02	1.48E−01	3.64E−03	0
	9.31	8.09	25.70	8.46	2.02	2.16	0.00

DOF: 9

Table 4 Statistical results for detectors on scaling (hypothesis index, followed by t value and SD)

	BRISK	CENSURE	FAST	MSER	ORB	SIFT	SURF
BRISK	0	0	0	0	0	1	1
	0	5.64E−01	9.92E−01	1.40E−01	2.76E−01	1.93E−02	4.13E−03
	0.00	12.08	10.89	8.03	16.76	13.14	15.59
CENSURE	0	0	0	1	0	1	1
	5.64E−01	0	6.96E−01	1.54E−02	3.12E−01	1.08E−02	1.23E−02
	12.08	0.00	18.25	6.80	24.90	13.93	21.37
FAST	0	0	0	0	0	0	1
	9.92E−01	6.96E−01	0	4.08E−01	1.33E−01	8.72E−02	7.45E−03
	10.89	18.25	0.00	14.86	11.70	19.42	17.27
MSER	0	1	0	0	0	1	1
	1.40E−01	1.54E−02	4.08E−01	0	7.81E−01	3.55E−02	1.90E−02
	8.03	6.80	14.86	0.00	22.40	9.86	16.27
ORB	0	0	0	0	0	0	0
	2.76E−01	3.12E−01	1.33E−01	7.81E−01	0	5.25E−01	1.24E−01
	16.76	24.90	11.70	22.40	0.00	27.12	23.59
SIFT	1	1	0	1	0	0	0
	1.93E−02	1.08E−02	8.72E−02	3.55E−02	5.25E−01	0	6.49E−02
	13.14	13.93	19.42	9.86	27.12	0.00	10.49
SURF	1	1	1	1	0	0	0
	4.13E−03	1.23E−02	7.45E−03	1.90E−02	1.24E−01	6.49E−02	0
	15.59	21.37	17.27	16.27	23.59	10.49	0.00

DOF: 9

Table 5 Statistical results for detectors on rotation (hypothesis index, followed by t value and SD)

	BRISK	CENSURE	FAST	MSER	ORB	SIFT	SURF
BRISK	0	1	1	1	1	1	1
	0	3.68E−03	9.17E−06	3.51E−06	5.60E−06	6.82E−06	6.10E−06
	0.00	29.86	1.65	10.12	7.78	6.51	3.29
CENSURE	1	0	1	0	1	0	1
	3.68E−03	0	7.12E−03	3.55E−01	2.95E−04	5.84E−02	1.22E−03
	29.86	0.00	29.42	26.88	33.25	27.83	31.47
FAST	1	1	0	1	1	1	1
	9.17E−06	7.12E−03	0	3.53E−06	4.53E−06	7.79E−06	4.37E−06
	1.65	29.42	0.00	8.62	9.16	4.97	4.72
MSER	1	0	1	0	1	1	1
	3.51E−06	3.55E−01	3.53E−06	0	3.36E−06	6.89E−06	3.57E−06
	10.12	26.88	8.62	0.00	17.47	4.29	13.26
ORB	1	1	1	1	0	1	1
	5.60E−06	2.95E−04	4.53E−06	3.36E−06	0	3.36E−06	2.78E−05
	7.78	33.25	9.16	17.47	0.00	13.47	5.32
SIFT	1	0	1	1	1	0	1
	6.82E−06	5.84E−02	7.79E−06	6.89E−06	3.36E−06	0	5.30E−06
	6.51	27.83	4.97	4.29	13.47	0.00	9.59
SURF	1	1	1	1	1	1	0
	6.10E−06	1.22E−03	4.37E−06	3.57E−06	2.78E−05	5.30E−06	0
	3.29	31.47	4.72	13.26	5.32	9.59	0.00

DOF: 12

Table 6 Statistical results for descriptors on viewpoint (hypothesis index, followed by t value and SD)

	BRIEF	BRISK	FREAK	ORB	SIFT	SURF
BRIEF	0	1	1	1	1	1
	0	3.05E−35	3.48E−24	1.88E−19	5.28E−30	1.26E−22
	0.00	1.47	1.55	0.84	1.16	5.03
BRISK	1	0	1	1	1	1
	3.05E−35	0	1.72E−13	2.71E−39	5.89E−07	2.51E−12
	1.47	0.00	1.22	1.65	2.25	4.42
FREAK	1	1	0	1	0	1
	3.48E−24	1.72E−13	0	6.52E−31	5.88E−01	5.68E−19
	1.55	1.22	0.00	1.70	2.34	3.98
ORB	1	1	1	0	1	1
	1.88E−19	2.71E−39	6.52E−31	0	1.80E−36	1.73E−25
	0.84	1.65	1.70	0.00	1.33	5.09
SIFT	1	1	0	1	0	1
	5.28E−30	5.89E−07	5.88E−01	1.80E−36	0	2.25E−12
	1.16	2.25	2.34	1.33	0.00	5.86
SURF	1	1	1	1	1	0
	1.26E−22	2.51E−12	5.68E−19	1.73E−25	2.25E−12	0
	5.03	4.42	3.98	5.09	5.86	0.00

DOF: 70

Table 7 Statistical results for descriptors on blurring (hypothesis index, followed by t value and SD)

	BRIEF	BRISK	FREAK	ORB	SIFT	SURF
BRIEF	0	0	0	0	0	0
	0	9.47E−01	6.47E−02	1.36E−01	1.93E−01	5.37E−01
	0.00	37.97	17.36	1.34	24.91	47.57
BRISK	0	0	0	0	0	0
	9.47E−01	0	3.14E−01	9.92E−01	6.30E−02	2.05E−01
	37.97	0.00	31.87	37.61	17.74	20.48
FREAK	0	0	0	0	1	0
	6.47E−02	3.14E−01	0	7.82E−02	2.30E−02	8.79E−01
	17.36	31.87	0.00	17.28	26.15	38.12
ORB	0	0	0	0	0	0
	1.36E−01	9.92E−01	7.82E−02	0	1.66E−01	5.65E−01
	1.34	37.61	17.28	0.00	24.69	47.47
SIFT	0	0	1	0	0	0
	1.93E−01	6.30E−02	2.30E−02	1.66E−01	0	6.13E−02
	24.91	17.74	26.15	24.69	0.00	30.69
SURF	0	0	0	0	0	0
	5.37E−01	2.05E−01	8.79E−01	5.65E−01	6.13E−02	0
	47.57	20.48	38.12	47.47	30.69	0.00

DOF: 9

3. The alternate hypothesis: for two methods, 1 and 2, the means are different, i.e., $\mu_1 \neq \mu_2$; thus, the methods perform differently.
4. The tests are conducted under 95 % confidence level for a two-tailed test.

If the null hypothesis is true with significant differences in results, this would signify that the experiment may be noisy. Thus, careful observations for the results are required. The results of the test are reported in Tables 2, 3, 4, 5, 6, 7, 8 and 9. Each cell in the tables contains three sub-rows under each row signifying the hypothesis, t value from the test, and the standard deviation (SD). The first row is either 0 (supporting the null hypothesis) or 1 (supporting the alternate hypothesis). Finally, the degree of freedom

Table 8 Statistical results for descriptors on scaling (hypothesis index, followed by t value and SD)

	BRIEF	BRISK	FREAK	ORB	SIFT	SURF
BRIEF	0	0	0	0	0	0
	0	7.47E−01	5.00E−01	7.44E−01	2.25E−01	6.64E−01
	0.00	13.17	13.35	8.78	16.66	17.75
BRISK	0	0	0	0	0	0
	7.47E−01	0	3.63E−01	6.10E−01	3.05E−01	7.71E−01
	13.17	0.00	14.36	13.87	24.01	12.05
FREAK	0	0	0	0	0	0
	5.00E−01	3.63E−01	0	2.99E−01	6.62E−01	1.16E−01
	13.35	14.36	0.00	5.81	27.31	9.97
ORB	0	0	0	0	0	0
	7.44E−01	6.10E−01	2.99E−01	0	4.35E−01	4.23E−01
	8.78	13.87	5.81	0.00	22.97	13.04
SIFT	0	0	0	0	0	0
	2.25E−01	3.05E−01	6.62E−01	4.35E−01	0	3.39E−01
	16.66	24.01	27.31	22.97	0.00	29.39
SURF	0	0	0	0	0	0
	6.64E−01	7.71E−01	1.16E−01	4.23E−01	3.39E−01	0
	17.75	12.05	9.97	13.04	29.39	0.00

DOF: 9

Table 9 Statistical results for descriptors on rotation (hypothesis index, followed by t value and SD)

	BRIEF	BRISK	FREAK	ORB	SIFT	SURF
BRIEF	0	1	1	0	1	1
	0	4.34E−06	7.26E−06	2.77E−01	9.55E−04	1.43E−02
	0.00	15.38	14.40	1.65	2.75	14.18
BRISK	1	0	1	1	1	1
	4.34E−06	0	1.90E−04	5.27E−06	4.02E−06	1.03E−04
	15.38	0.00	2.53	15.43	16.77	14.22
FREAK	1	1	0	1	1	1
	7.26E−06	1.90E−04	0	9.85E−06	5.81E−06	1.04E−04
	14.40	2.53	0.00	14.58	15.64	11.88
ORB	0	1	1	0	1	1
	2.77E−01	5.27E−06	9.85E−06	0	5.63E−03	2.11E−02
	1.65	15.43	14.58	0.00	4.11	14.60
SIFT	1	1	1	1	0	1
	9.55E−04	4.02E−06	5.81E−06	5.63E−03	0	4.31E−03
	2.75	16.77	15.64	4.11	0.00	14.97
SURF	1	1	1	1	1	0
	1.43E−02	1.03E−04	1.04E−04	2.11E−02	4.31E−03	0
	14.18	14.22	11.88	14.60	14.97	0.00

DOF: 12

(DOF) is reported in the caption of each table. This is a fixed number depending on the number of samples on which the t test is performed. For the experiments, the number of samples represents the number of transformation values and the DOF is one less than the number of samples.

A thorough explanation of the tables requires considerable space. Instead, we report select observations:

1. Diagonal elements for all tables have zero values. This is evident, as statistically both distributions represent the same pair and are identical.
2. The t values are tested from the available Student's t test charts for 95 % confidence level under the given DOF.
3. A value of 1 for the hypothesis represents the alternate hypothesis. As presented in Table 2, most of the combinations received a value of 1. This signifies that, for

a given descriptor (here, it is BRIEF), the performances of two detectors for a viewpoint transformation are statistically different. This also proves that the variations of results produced by the tests mostly denote variations in performance and not simply noise. As can be seen in Table 2, the performance of CENSURE and MSER, as well as CENSURE and SIFT, are similar. This also supports the results presented in Fig. 4 (left column-top row) as CENSURE, MSER, and SIFT are the three best performers, while from Fig. 3b we can see that their mean precision values are very close to each other. As evident in Table 6, most of the descriptors have performed statistically different for viewpoint changes when the detector is kept constant (here, it is FAST), apart from SIFT and FREAK. This is to be expected, since their performance on viewpoint changes (Fig. 4 and in Fig. 3b) are similar.

4. As seen in Table 3, most of the detectors have statistically different performances in the case of blurring, as verified by the majority of 1 s. However, ORB is similar to SURF and BRISK. ORB was proposed as a faster alternative to SURF, and its performance is close to that of SURF (from Fig. 3d). However, the similarity of BRISK to ORB and SURF is surprising. Similarities between SIFT to MSER can also be observed, attributable to the fact that the extractions of feature points for both detectors are similar. As seen in Table 7, most descriptor performances are statistically similar in the case of blurring. This is because descriptors are not generally responsible for invariance against blurring. This can also be verified from results presented in Fig. 4, where the performances of the descriptors are somewhat arbitrary in nature.
5. In Table 4, we observe that a number of detectors have similar statistical performances. This is because some of the detectors are more resistant to scaling than others. For example, SIFT and SURF are scale invariant and their performances are similar, while FAST is not scale invariant and its performance is different from that of SIFT or SURF. However, as stated above, as BRIEF is a descriptor of moderate quality, these results also contain some noise. All descriptors in the presence of scaling have similar performances, as shown in Table 8. This is also expected, since, like in the case of blur, invariance against scaling is mostly achieved at the detection level. Thus, descriptors have a little role to play in the case of invariance against scaling.
6. Finally, for rotational invariance testing, most of the detectors and descriptors have statistically different results, as shown in Tables 5 and 9. Although, rotational invariance is mostly achieved in the stage of descriptor assignment, it partly depends on detectors due to the selection of feature points. Thus, statistically different performances are expected.

4.5 Dependence on parameters

In reference to the metric values in Fig. 3, a mismatch can be noticed. CENSURE provides high Precision values, while its Reliability is low. The reason for such odd behavior cannot be subtly judged from the experiments. Some detectors and descriptors provide good results on some images, while they do not work well with others. This observation leads to an assumption that detectors and descriptors are more comfortable within specific transformation value ranges and for certain type of image contents. While judgement based on image content would be a qualitative assessment that is beyond the scope of this work, reaction to the change of transformation values can be measured.

In response to such confusion, some of the parameters related to detection for BRISK, FAST, and CENSURE are adjusted, and matching performance is judged on the ALOI viewpoint dataset. The viewpoint dataset is chosen as it has naturally occurring viewpoint transformations with many transformation values, improving the robustness of the experiment. Variation of Precision values with changes in parameter values are shown in Fig. 5. The parameters used in the experiment are as follows:

BRISK:

- Octave: the octave of the scale-space pyramid used. Each octave represents the progressive half-sampled image pyramid. Higher octaves signify better detection.
- Threshold: this is the detection threshold used for BRISK. More specifically, it is the threshold used for the initial detector in BRISK: the FAST 9–16 detector. Thus, a higher threshold value would indicate fewer detected points and fewer incorrect detections.

FAST:

- Threshold: the threshold is used to detect corner pixels using the accelerated segment test. A high threshold value indicates stricter criteria for detecting corners in a neighborhood of pixels and fewer incorrect detections.

CENSURE:

- Line threshold: line threshold is a ratio for the principal curvatures that decide the number of features to be considered along an edge or a line. These features are often improperly localized, and hence perform poorly. In the implementation by OpenCV, we use two line thresholds (lineThresholdProjected for Harris measure of response and lineThresholdBinarized for Harris measure of sizes). Any response along the edges that satisfies both threshold comparisons is considered to be acceptable. For the test, lineThresholdProjected is varied while lineThresholdBinarized is taken as 0.8 of lineThresholdProjected.

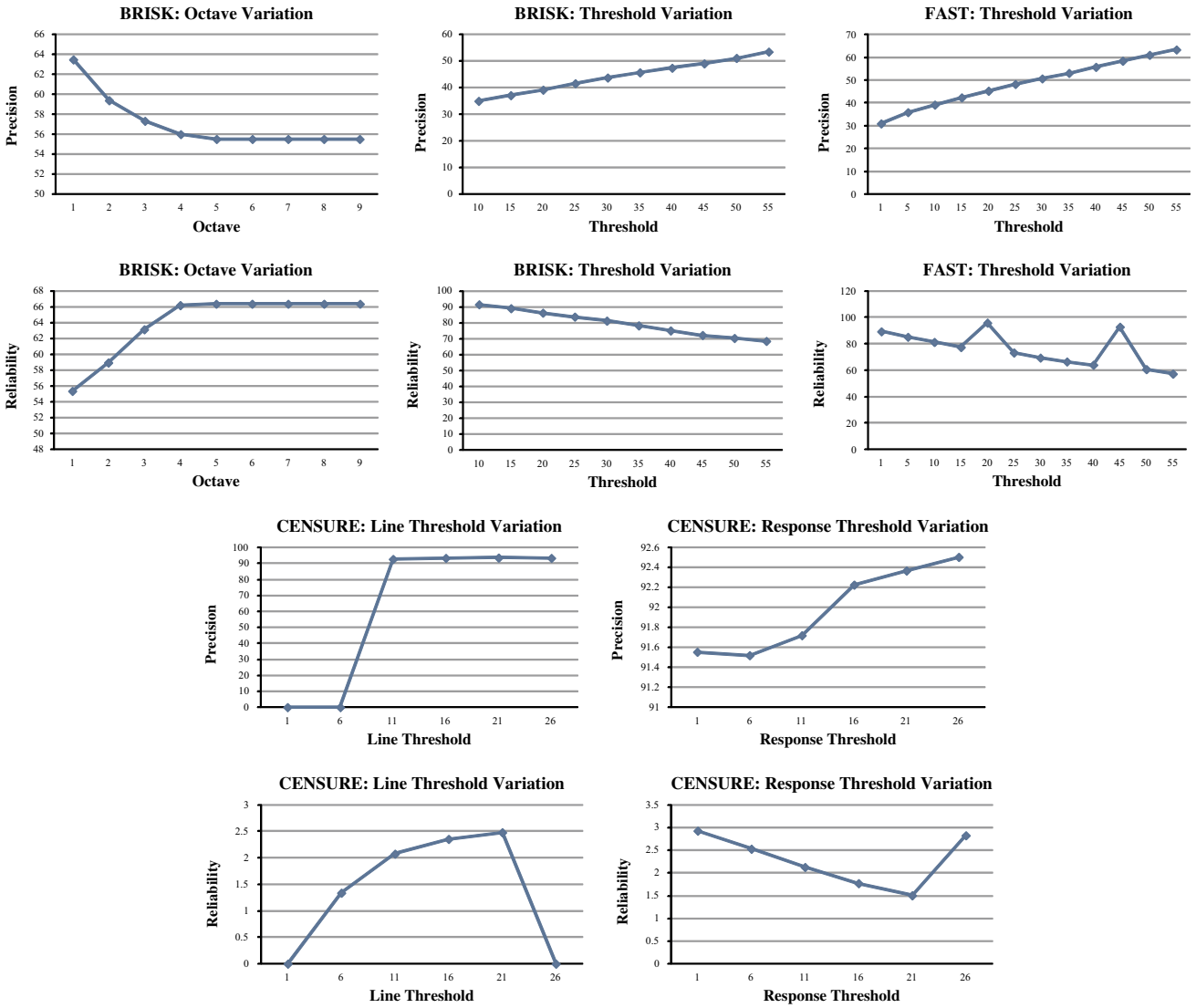

Fig. 5 Precision and Reliability variation with parameter changes are shown for BRISK, FAST, and CENSURE

Obviously, an increase in threshold would lead to fewer features along edges/lines.

- Response threshold: in CENSURE, seven filters are applied to images and the responses obtained are processed with non-maximal suppression. Weak corner responses are removed with the response threshold. Thus, a high threshold would keep only the better features.

Observations:

1. For most of the experiments, Precision and Reliability follow an inverse relationship: an increase in Precision relates to a decrease in Reliability. This phenomenon can also be observed in previous experiments. This phenomenon occurs, because an increase in Precision is the result of the removal of unstable and confusing matches. However, removing many matches leads to no matches for some of the images, resulting in decreased Reliability.
2. The parameter variations provide an easy way to find an optimal parameter value for which Precision and Reliability have desired values. For example, an octave value of 4 would provide high Reliability with moderate Precision for BRISK.
3. The values close to 26 yield a peculiar nature of CENSURE. A line threshold greater than 26 drastically reduces the Reliability, while a high response threshold of 26 increases both the Precision and Reliability. This phenomenon is partially supported in [11].
4. Finally, there are two peaks in the Reliability for FAST. High thresholds generally prevent detection of feature points and increase precision. As discussed, this should reduce the Reliability as well. However, this situation may be a result of the fundamental way FAST works.

Table 10 Mean execution time(s) for each pair of detector and descriptor (in seconds)

	BRIEF	BRISK	FREAK	ORB	SIFT	SURF
BRISK	0.26	0.30	0.29	0.21	17.94	2.13
CENSURE	0.37	0.34	0.36	0.32	3.77	0.74
FAST	0.25	0.25	0.26	0.19	2.17	0.51
MSER	1.21	1.26	1.47	1.14	16.46	2.93
ORB	0.26	0.22	0.33	0.20	46.62	4.24
SIFT	0.57	0.56	0.59	0.55	3.03	0.92
SURF	0.62	0.78	0.75	0.60	14.99	2.51

Table 11 Average ranks for the compared combinations of detectors and descriptors across all transformations

	BRIEF	BRISK	FREAK	ORB	SIFT	SURF
BRISK	10	6	4	15	16	17
CENSURE	7	10	11	8	3	23
FAST	2	6	5	7	1	23
MSER	17	10	11	19	18	21
ORB	17	10	9	13	17	14
SIFT	19	22	20	12	8	24
SURF	15	11	10	15	16	21

The threshold is used to compare each surrounding pixel to the center pixel to decide, based on an entropy, whether the center pixel is a corner or a regular pixel. Based on the threshold, the surrounding pixels are put into three groups: darker, similar, or brighter. The entropy relation is based on the relation of the center pixel to the surrounding pixels. Thus, the way the surrounding pixel is related can either increase or decrease the entropy and mark the center pixel as a corner or a regular pixel. For the test images in this study, threshold values of 20 and 45 possibly provide more information for the center pixel and improve the performance. Of course, the performance depends on the illumination variations of the images. For ALOI, the center object is highly illuminated while the outside pixels are totally dark. It is possible that if other datasets are used, the Reliability graph may not have the same peaks.

An extensive study comparing the parameters of each technique can be a very interesting independent subject. The variation of the Precision and Reliability for different types of transformation may also be useful for related research. Due to the limited space of this work, we conclude the parameter variation experiment here and keep these additional studies for future work.

4.6 Rank computation

The main objective for undertaking such extensive experimentation is to compare the feature detection and/or description techniques. Due to the unavailability of detectors or descriptors for a single technique, comparisons cannot take place by straight-forward comparison on the performance of each technique. Instead, comparisons are conducted for each pair of detectors and descriptors, and the results are tabulated in Tables 10 and 11. The term "method" represents a combination of a detector and a descriptor, similar to Sect. 4.4. For comparison, five types of metrics are available: Putative Match Ratio, Precision, Matching Score, Reliability, and Mean Execution Time. The Mean Execution Times are listed in Table 10. Individual metric ranks for each transformation are easy to compute. However, a single rank based on all metrics over all transformations would be beneficial.

In view of the above, a ranking methodology has been adopted from Goyette et al. [17]. Although it was developed for change detection, we have found the methodology equally useful for ranking detectors and descriptors based on metric values on transformations. The procedure for ranking is provided below:

1. For method m (representing a combination of a detector and a descriptor) with transformation value e and metric t, find the positional rank $\mathbb{P}_{m,e,t}$ by sorting the metric values (a high value indicates better positional rank for the Putative Match Ratio, Precision, Matching Score, and Reliability while a low value indicates better positional rank for mean execution time).
2. Find the average rank of m for transformation e, over all metrics: $\mathbb{R}_{m,e} = \frac{1}{N_t} \sum_t \mathbb{P}_{m,e,t}$. Here, N_t represents the number of metrics.
3. Also, compute the average rank of m for metric t, over all transformations: $\mathbb{R}_{m,t} = \frac{1}{N_e} \sum_e \mathbb{P}_{m,e,t}$. Here, N_e represents the number of transformations.
4. Determine the positional rank of each method m for each transformation e based on the value of $\mathbb{R}_{m,e}$, by sorting $\mathbb{R}_{m,e}$ in ascending order (low $\mathbb{R}_{m,e}$ indicates better positional rank). We do not use simple position in assigning positional ranks. instead, the same positional rank is assigned if two subsequent ranks have very similar values within a certain threshold T. Thus, the positional ranks begin with 1 and increase only when two subsequent ranks have a difference of more than T. The positional rank is represented as $\mathbb{P}_{m,e}$. Similarly, positional ranks from $\mathbb{R}_{m,t}$ are represented as: $\mathbb{P}_{m,t}$. The choice of T requires explanation. T cannot be a constant value, as the variance of distribution for $\mathbb{R}_{m,e}$ or $\mathbb{R}_{m,t}$ may not be equal. Thus, two values may be far apart yet similar for one of the distributions, while two very close values may require separation for the other distribution. Thus, T needs to depend on the variance and, specifically, the SD of the distribution. For the rankings, we

choose $T = 0.05 \times \text{SD}$. A smaller multiplication factor is chosen so as to satisfy a stringent similarity criteria. With higher values of the factor, more methods will be assigned similar ranks while lower values would separate them. Thus, the choice would also depend on the application. Here, we show the effect for only one value due to limited space.

5. Find the average rank across transformations: $\mathbb{R}_m = \frac{1}{N_e} \sum_e \mathbb{P}_{m,e}$. Here, N_e represents the number of transformations.
6. Finally, determine $\mathbb{P}_m$, the positional rank across transformations by sorting $\mathbb{R}_m$ in ascending order.

To evaluate the methods, the positional average ranks $\mathbb{P}_{m,t}$ for each metric across all transformations, and average positional ranks $\mathbb{P}_m$ across all transformations are provided in Fig. 3j and Table 11, respectively. According to Table 11, FAST+SIFT and FAST+BRIEF are ranked on top. This is strange for FAST+BRIEF, because neither FAST nor BRIEF has good invariance against higher degree of transformations. However, according to the consolidated results in Fig. 3, although their ranks are not the best for blurring, scaling, or rotation in terms of Precision, their combination has achieved a higher level of consistency for most of the transformations and for the rest of the metrics, accounting for their ranking. FAST+FREAK has also performed well. This leads to the observation that the feature detector makes a significant contribution to the performance of the pair. It may not be clear from Fig. 4 (first column) that how the descriptor limits the performance of the detectors, as the descriptor is responsible for extracting highly invariant features from detected feature points; however, a detector that detects appropriate feature points leads to a better detection. FAST detects a lot of feature points, out of which, some are stable and some are not. As the number of stable feature points is also high, it leads to a better overall performance. It can also be noted in Table 11 that a column summation would lead to a detector rank over all descriptors, and a row summation would lead to a descriptor rank over all detectors. Finally, the ranking can be made different by weighing specific metrics and transformations while taking the averages. Future work on ranking for specific applications and based on specialized metrics may prove beneficial.

4.7 A practical scenario

This section reports a comparison of the combinations of detectors and descriptors in a practical scenario. To limit the amount of experimentation, only a few pairs of detectors and descriptors are used. We have specifically used the unified pairs intuitively chosen in Sect. 4.3. Other choices are equally applicable, and readers are encouraged to experiment with other pairs. For this experiment, we have used the Kitti Visual Odometry datasets developed by Geiger et al. [12]. The database information is briefly described below:

- Our goal is to extract the motion of a moving vehicle using stereo images and recording the visual odometry.
- A standard station wagon is equipped with two high-resolution color and grayscale video cameras to capture images at a fixed interval.
- Stereo image datasets are captured by driving the vehicle around the city of Karlsruhe, in rural areas and on highways.
- Accurate ground truth positional data are captured using a Velodyne laser scanner and a GPS localization system.
- The camera calibration is provided for each dataset.
- The first 11 sequences of the datasets have ground truth odometry information. A total of 23,201 stereo frames are used for the experiment.
- The datasets contain different image sizes (1241×376), (1242×375), and (1226×370). More information can be found on the referred web site.

The experiment is described in the following steps:

1. Detect feature points and extract descriptors in each stereo pair.
2. Match the feature points using the brute-force method, as in Experiment I.
3. Estimate the ego motion from the stereo pairs in current time instances and in previous time instances, using the five-step approach defined by Kitt et al. [24].
4. Match the estimated positions of the vehicle to the ground truth to evaluate the performance of the compared pairs of detectors and descriptors.

According to the above, the experiments are divided into four steps. Step 1 corresponds to the main comparison of the pairs of detectors and descriptors. Step 2 is similar to the matching procedure used in Experiment I. Steps 3 and 4 require discussion. Kitt et al. [24] have designed a visual odometry estimation procedure based on stereo image sequences with uniform bucketing of image features, the RANSAC-based outlier rejection scheme, and an iterated sigma point Kalman filter-based ego motion extraction. They also provide an implementation of the procedure on their web site. We use their implementation for accurate results for their datasets. The implementation details of the approach are described in the reference and are not repeated here.

In regard to step 1 of the experiment, sparse matching results on stereo image pairs from the dataset are provided in Fig. 6. As can be observed from the figure, the num-

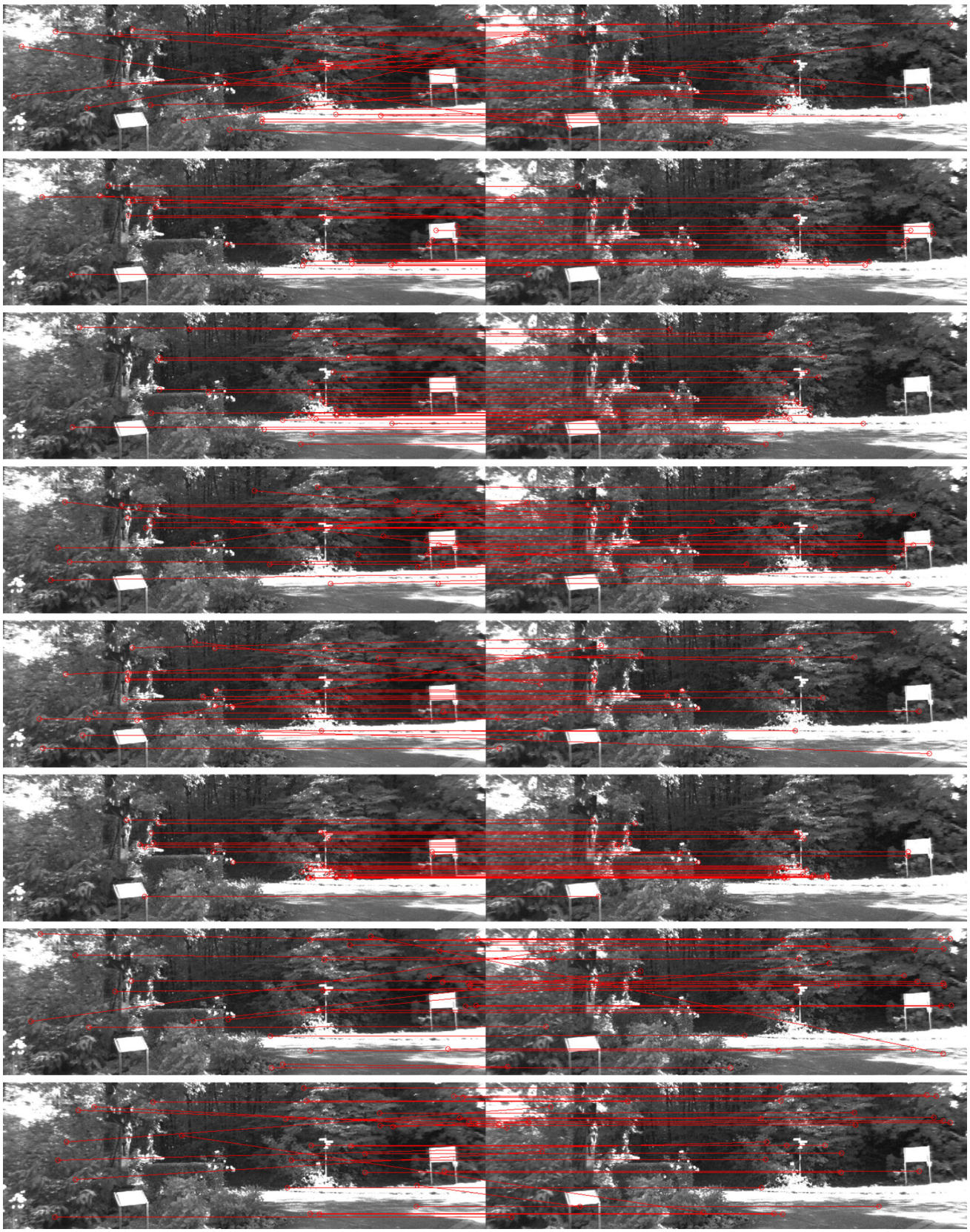

Fig. 6 Sparse matching results on a Kitti dataset image from sequence 03. The *rows* from the *top* represent the matching results for BRISK, CENSURE+BRIEF, FAST+BRIEF, SURF+FREAK, MSER+BRIEF, ORB, SIFT, and SURF, respectively

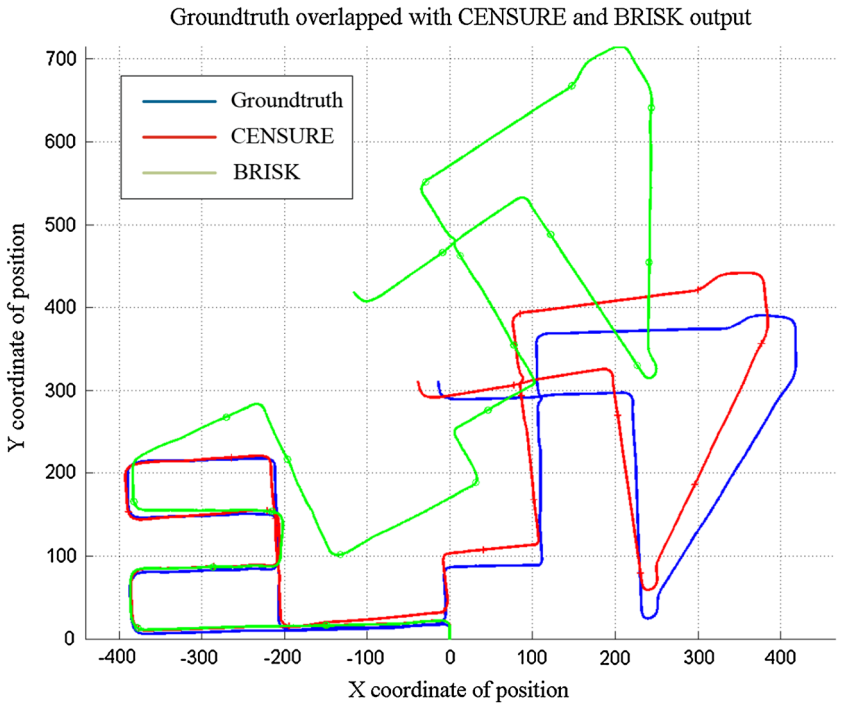

Fig. 7 Output egomotion of sequence 08 from the Kitti dataset. The *plot* provides the x (*horizontal* motion in camera plane) and z (motion towards camera axis) values of the camera ego motion. However, the plotting has been conducted against the X–Y axis with Y representing the z (forward) motion. The ground truth motion is overlapped with the output from CENSURE+BRIEF and BRISK (color figure online)

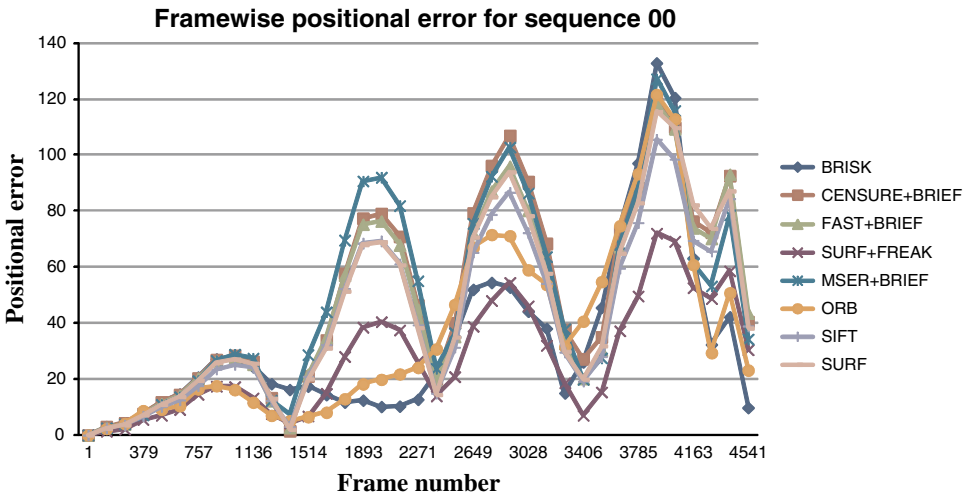

Fig. 8 The comparison of framewise positional error (log of Euclidean distance)

ber of detected features depends on the complexity of the scene and the performance of the algorithm. Thus, a fair comparison can be made of the performance when a common scene is used. Following steps 1–4, the positions of the vehicle for each stereo pair can be estimated. The positions are found in meters with respect to the starting position, i.e., the first stereo pair. The positions can be plotted on an X–Y plane for visual inspection. Such a positional plot for sequence 08 has been displayed in Fig. 7. In the figure, the ground truth data (in blue) have been overlapped with the data from CENSURE+BRIEF (red), and BRISK (green). The starting position is (0,0). CENSURE+BRIEF and BRISK are taken as the two pairs for the positional error extremas. As can be seen from Fig. 7, the output from BRISK starts to deviate more from the ground truth compared to CENSURE+BRIEF. In the end, the difference is large due to the accumulation of errors for consecutive stereo pairs.

The error needs to be quantified to compare the odometry results with the ground truth data. The error metric used for the experiment is given in Eq. 1.

$$\mathrm{err}(f) = \log\left(\sqrt{((gt(f,1)-r(f,1))^2+(gt(f,2)-r(f,2))^2)}\right). \tag{1}$$

Table 12 Average positional error for all sequences and the overall average

Methods	00	01	02	03	04	05	06	07	08	09	10	Overall average
BRISK	34.14	407.32	288.46	8.71	2.18	43.77	8.98	9.14	158.71	150.72	136.20	113.49
CENSURE+BRIEF	50.07	233.92	49.80	3.65	2.03	12.86	10.95	4.94	22.12	16.40	5.64	37.49
FAST+BRIEF	47.43	145.73	67.61	7.30	1.65	16.32	7.35	5.82	34.54	24.79	3.37	32.90
SURF+FREAK	27.54	162.48	56.67	4.67	3.11	11.54	7.22	4.56	28.02	17.21	3.10	29.65
MSER+BRIEF	49.95	710.86	41.48	5.58	2.55	44.64	14.67	9.01	38.95	147.95	24.99	99.15
ORB	36.52	307.70	53.68	11.15	6.96	41.54	20.31	8.09	52.04	65.72	11.02	55.88
SIFT	45.73	131.38	74.36	5.70	3.30	15.20	7.29	7.07	36.33	20.50	5.78	32.06
SURF	43.00	170.99	57.92	4.91	3.00	14.20	7.92	6.51	35.51	18.61	3.12	33.24

Table 13 Comparison of computational time for motion recovery

Methods	Time for 50 frames	Average time/frame
BRISK	87.98	1.76
CENSURE+BRIEF	23.78	0.47
FAST+BRIEF	74.46	1.49
SURF+FREAK	180.07	3.60
MSER+BRIEF	52.20	1.04
ORB	16.07	0.32
SIFT	172.93	3.46
SURF	791.45	15.83

Here, err(f) is the error output for frame f. gt and r contain the ground truth and resulting (x, y) position of the camera, respectively. $gt(f, 1)$ and $gt(f, 2)$ represent the x and z values for frame f, respectively. This is similar for $r(f, 1)$ and $r(f, 2)$. The logarithm has been taken to reduce the differences between errors, as some of the pairs of detectors and descriptors have large ranges of errors. The positional error and its deviation can be clearer with a plot of the error.

A plot of the positional error for sequence 00 is shown in Fig. 8. The x-axis is the frame number, while the y-axis represents err(f), which is the logarithm of the Euclidean distance in meters. To simplify the label notation, it is shown as Positional Error. As evident from Fig. 8, SURF+FREAK performs better than others with a low positional error, while CENSURE+BRIEF and MSER+BRIEF have high ranges of positional errors.

The average error per sequence and the average over all sequences have been tabulated in Table 12. The combined average shows that SURF+FREAK performs best among the pairs, while BRISK has the highest error rate. The performance cannot be properly judged based on a single sequence, because the rates and ranks of the pairs vary considerably with different types of images. However, a cumulative judgement gives a better picture. Also, some pairs exhibit higher positional errors. This is due to their inability to provide accurate matching results for some of the frames. As the ego motion is calculated based on the values of previous frame, a wrong calculation on one frame can affect the calculations on subsequent frames, resulting in higher cumulative positional errors.

Finally, the computational time for 50 frames from sequence 3 with an image resolution of (1242 × 375) has been tabulated in Table 13, where the comparison framework is implemented using C++ and run on a desktop PC with 3.4 GHz Intel Core i7-3770 CPU. The table provides another side of the experiment. Although FREAK performs best among the descriptors, computationally it does not yield the benefit of speed, while ORB and CENSURE+BRIEF provide better choices.

5 Conclusion

The paper has conducted an extensive comparative study of nine popular feature detectors and descriptors. The study has made the following contributions: (1) the experiments provide a broader idea of the performance of feature detectors and descriptors against several image transformations like blurring, rotation, scaling, and viewpoint changes in varying degrees; (2) cumulative comparisons of several transformations over a number of databases have been performed for a total of 42 combinations of detectors and descriptors; (3) a study on parameters has been carried out for three detectors; (4) a statistical analysis has been conducted separately for detectors and descriptors for four major transformations; (5) a sparse matching-based application framework has been established to verify the performances of some combinations of detectors and descriptors in practical situations; and (6) the rankings shown in the paper highlight the relative performances of the detectors and descriptors and can be made more application specific by adapting to suitable weighing methods and thresholds.

Several observations were made during the study. Some of the observations are new, while others conform to the previous studies. For example, the performance of BRIEF as a descriptor has been properly established through a number of experiments and also the overall ranking. This also resonates with the works of Heinly et. al. [20]. Also, SIFT has shown the best performance with its own descriptor. How-

ever, its descriptor has shown great performance when paired up with the FAST detector. This is one of the new observations coming out from the study. Finally, ORB has shown a reliable performance for most of the experiments, and hence, is quite suitable for a number of vision related applications.

The last part of the work shows a practical evaluation of feature detectors and descriptors in feature tracking. While this study is beneficial for several practical scenarios, it also highlights Reliability as one of the predominant challenges of feature tracking. Minor loss of reliable tracking for a few consequent frames can result in a large positional error. While there are a number of methodologies available to recover from such errors, such as Kalman filtering, or probabilistic modeling, a reliable detector and descriptor combination can improve the tracking accuracy by a large margin and restrain the use of such additional methods. Thus, a measure like Reliability is required for feature tracking.

Future work will include experiments with a broader variation of images, the inclusion of more feature detectors, and a study on parameters for each detector and descriptor.

Acknowledgments The authors would like to thank the anonymous reviewers for their helpful and constructive comments. We would like to thank Geiger et al. [13] for part of the code from LibViso2. The work is supported in part by the Canada Research Chair program, AUTO21 Networks of Centres of Excellence, and the Natural Sciences and Engineering Research Council of Canada.

References

1. Aanæs, H., Dahl, A., Steenstrup Pedersen, K.: Interesting interest points. Int. J. Comput. Vis. **97**, 18–35 (2012)
2. Agrawal, M., Konolige, K., Blas, M.: CenSurE: center surround extremas for realtime feature detection and matching. In: Proceedings of European Conference on Computer Vision, pp. 102–115 (2008)
3. Alahi, A., Ortiz, R., Vandergheynst, P.: Freak: Fast retina keypoint. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 510–517 (2012)
4. Bay, H., Tuytelaars, T., Gool, L.V.: SURF: speeded up robust features. In: Proceedings of the European Conference on Computer Vision, pp. 404–417 (2006)
5. Calonder, M., Lepetit, V., Strecha, C., Fua, P.: BRIEF: binary robust independent elementary features. In: Proceedings of the European Conference on Computer Vision, pp. 778–792 (2010)
6. Dahl, A.L., Aanæs, H., Pedersen, K.S.: Finding the best feature detector–descriptor combination. In: International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), pp. 318–325 (2011)
7. Feng, Y., Ren, J., Jiang, J., Halvey, M., Jose, J.: Effective venue image retrieval using robust feature extraction and model constrained matching for mobile robot localization. Mach. Vis. Appl. **23**(5), 1011–1027 (2012)
8. Fernández, A., Ghita, O., González, E., Bianconi, F., Whelan, P.: Evaluation of robustness against rotation of LBP, CCR and ILBP features in granite texture classification. Mach. Vis. Appl. **22**(6), 913–926 (2011)
9. Forstner, W.: A framework for low level feature extraction. In: Proceedings of the European Conference on Computer Vision, pp. 383–394 (1994)
10. Gao, J., Huang, X., Liu, B.: A quick scale-invariant interest point detecting approach. Mach. Vis. Appl. **21**(3), 351–364 (2010)
11. Gauglitz, S., Höllerer, T., Turk, M.: Dataset and evaluation of interest point detectors for visual tracking. Technical Report, Department of Computer Science, University of California (2010)
12. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: The Kitti vision benchmark suite. http://www.cvlibs.net/datasets/kitti/eval_odometry.php (2002). Accessed 12 March 2013
13. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? The Kitti vision benchmark suite. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (2012)
14. Geusebroek, J.M., Burghouts, G., Smeulders, A.M.: The Amsterdam library of object images. Int. J. Comput. Vis. **61**(1), 103–112 (2005)
15. Gil, A., Mozos, O., Ballesta, M., Reinoso, O.: A comparative evaluation of interest point detectors and local descriptors for visual slam. Mach. Vis. Appl. **21**(6), 905–920 (2010)
16. Govender, N.: Evaluation of feature detection algorithms for structure from motion. In: 3rd Robotics and Mechatronics Symposium (ROBMECH), pp. 1–4 (2009)
17. Goyette, N., Jodoin, P., Porikli, F., Konrad, J., Ishwar, P.: Changedetection.net: a new change detection benchmark dataset. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 1–8 (2012)
18. Hall, D., Leibe, B., Schiele, B.: Saliency of interest points under scale changes. In: British Mach. Vis. Conf., pp. 646–655 (2002)
19. Harris, C., Stephens, M.: A combined corner and edge detector. In: Proceedings of Alvey Vision Conference, pp. 147–151 (1988)
20. Heinly, J., Dunn, E., Frahm, J.M.: Comparative evaluation of binary features. In: Proceedings of European Conference on Computer Vision, pp. 759–773 (2012)
21. Heitger, F., Rosenthaler, L., von der Heydt, R., Peterhans, E., Kuebler, O.: Simulation of neural contour mechanism: from simple to end-stopped cells. Vis. Res. **32**(5), 963–981 (1992)
22. Kaneva, B., Torralba, A., Freeman, W.T.: Evaluating image feaures using a photorealistic virtual world. In: Proceedings of IEEE International Conference on Computer Vision (2011)
23. Khvedchenia, I.: Comparison of the opencv feature detection algorithms. http://computer-vision-talks.com/2011/07/comparison-of-the-opencvs-feature-detection-algorithms-ii/ (2011). Accessed 2 Nov 2012
24. Kitt, B., Geiger, A., Lategahn, H.: Visual odometry based on stereo image sequences with RANSAC-based outlier rejection scheme. In: IEEE Intelligent Vehicles Symposium (IV), pp. 486–492 (2010)
25. Leutenegger, S., Chli, M., Siegwart, R.: BRISK: binary robust invariant scalable keypoints. In: Proceedings of IEEE International Conference on Computer Vision, pp. 2548–2555 (2011)
26. Li, J., Allinson, N.M.: A comprehensive review of current local features for computer vision. Neurocomputing **71**(10–12), 1771–1787 (2008)
27. Liao, K., Liu, G., Hui, Y.: An improvement to the SIFT descriptor for image representation and matching. Pattern Recognit. Lett. **34**(11), 1211–1220 (2013)
28. Lindeberg, T.: Feature detection with automatic scale selection. Int. J. Comput. Vis. **30**(2), 79–116 (1998)
29. Lowe, D.: Object recognition from local scale-invariant features. In: Proceedings of IEEE International Conference on Computer Vision, vol. 2, pp. 1150–1157 (1999)
30. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide-baseline stereo from maximally stable extremal regions. Image Vis. Comput. **22**(10), 761–767 (2004)
31. Mikolajczyk, K., Schmid, C.: Indexing based on scale invariant interest points. In: Proceedings of IEEE International Conference on Computer Vision, pp. 525–531 (2001)

32. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Trans. Pattern Anal. Mach. Intell. **27**(10), 1615–1630 (2005)
33. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Gool, L.V.: A comparison of affine region detectors. Int. J. Comput. Vis. **65**(1–2), 43–72 (2005)
34. Moravec, H.P.: Towards automatic visual obstacle avoidance. In: Proceedings of International Joint Conference on Artificial Intelligence, p. 584 (1977)
35. Moreels, P., Perona, P.: Evaluation of features detectors and descriptors based on 3d objects. In: Proceedings of IEEE International Conference on Computer Vision, vol. 1, pp. 800–807 (2005)
36. Nistér, D., Stewénius, H.: Linear time maximally stable extremal regions. In: Proceedings of European Conference on Computer Vision, pp. 183–196 (2008)
37. Rosten, E., Drummond, T.: Machine learning for high-speed corner detection. In: Proceedings of European Conference on Computer Vision, pp. 430–443 (2006)
38. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: ORB: an efficient alternative to SIFT or SURF. In: Proceedings of IEEE International Conference on Computer Vision, pp. 2564–2571 (2011)
39. Schmid, C., Mohr, R., Bauckhage, C.: Comparing and evaluating interest points. In: Proceedings of IEEE International Conference on Computer Vision, pp. 230–235 (1998)
40. Schmid, C., Mohr, R., Bauckhage, C.: Evaluation of interest point detectors. Int. J. Comput. Vis. **37**(2), 151–172 (2000)
41. Shi, J., Tomasi, C.: Good features to track. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 593–600 (1994)
42. Smith, S.M., Brady, J.M.: SUSAN—a new approach to low level image processing. Int. J. Comput. Vis. **23**(1), 45–78 (1995)
43. Strecha, C., Von Hansen, W., Van Gool, L., Fua, P., Thoennessen, U.: On benchmarking camera calibration and multi-view stereo for high resolution imagery. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2008)
44. Tomasi, C., Kanade, T.: Detection and tracking of point features. Tech. rep., Int. Jnl. of Comput. Vision, Carnegie Mellon, Tech. Rep. (1991)
45. Tuytelaars, T., Mikolajczyk, K.: Local invariant feature detectors: a survey. Found. Trends Comput. Graph. Vis. **3**(3), 177–280 (2008)
46. USC-SIPI: the usc-sipi image database. http://sipi.usc.edu/database/database.php(1977). Accessed 29 Nov 2012
47. Ziegler, A., Christiansen, E., Kriegman, D., Belongie, S.: Locally uniform comparison image descriptor. In: Neural Info. Proc. Sys., pp. 1–9 (2012)
48. Zuliani, M., Kennedy, C., Manjunath, B.: A mathematical comparison of point detectors. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, vol. 11, pp. 172–178 (2004)

Dibyendu Mukherjee received his Bachelors Degree in Electronics and Telecommunications Engineering from Bengal Engineering and Science University, India in 2006. From 2006 to 2008, he worked as a consultant in PricewaterhouseCoopers Pvt. Ltd. He has completed his M.A.Sc degree from University of Windsor, Canada in the department of Electrical and Computer Engineering. He has recently completed his PhD from the same department. His research interests include image and video segmentation, stereo correspondence analysis, 3D Reconstruction, robot localization and tracking.

Q. M. Jonathan Wu received the Ph.D. degree in electrical engineering from the University of Wales, Swansea, U.K., in 1990. He was with the National Research Council of Canada for ten years from 1995, where he became a Senior Research Officer and a Group Leader. He is currently a Professor with the Department of Electrical and Computer Engineering, University of Windsor, Windsor, ON, Canada. He has published more than 250 peer-reviewed papers in computer vision, image processing, intelligent systems, robotics, and integrated microsystems. His current research interests include 3-D computer vision, active video object tracking and extraction, interactive multimedia, sensor analysis and fusion, and visual sensor networks. Dr. Wu holds the Tier 1 Canada Research Chair in Automotive Sensors and Information Systems. He is an Associate Editor for the IEEE Transactions on Systems, Man, and Cybernetics Part A, and the International Journal of Robotics and Automation. He has served on technical program committees and international advisory committees for many prestigious conferences.

Guanghui Wang is currently an assistant professor of Electrical Engineering and Computer Science at the University of Kansas, USA. His research interests include computer vision, image processing, and robotics. He has published one book at Springer-Verlag, and 70 papers in peer-reviewed journals and conferences. He served as associate editor or on the editorial board of five journals, and as an area chair or TPC member of 20+ conferences.