# Comparative Evaluation of Binary Features

Jared Heinly, Enrique Dunn, and Jan-Michael Frahm

The University of North Carolina at Chapel Hill
{jheinly,dunn,jmf}@cs.unc.edu

**Abstract.** Performance evaluation of salient features has a long-standing tradition in computer vision. In this paper, we fill the gap of evaluation for the recent wave of binary feature descriptors, which aim to provide robustness while achieving high computational efficiency. We use established metrics to embed our assessment into the body of existing evaluations, allowing us to provide a novel taxonomy unifying both traditional and novel binary features. Moreover, we analyze the performance of different detector and descriptor pairings, which are often used in practice but have been infrequently analyzed. Additionally, we complement existing datasets with novel data testing for illumination change, pure camera rotation, pure scale change, and the variety present in photo-collections. Our performance analysis clearly demonstrates the power of the new class of features. To benefit the community, we also provide a website for the automatic testing of new description methods using our provided metrics and datasets (www.cs.unc.edu/feature-evaluation).

**Keywords:** binary features, comparison, evaluation.

## 1 Introduction

Large-scale image registration and recognition in computer vision has led to an explosion in the amount of data being processed in simultaneous localization and mapping [1], reconstruction from photo-collections [2,3], object recognition [4], and panorama stitching [5] applications. With the increasing amount of data in these applications, the complexity of robust features becomes a hindrance. For instance, storing high dimensional descriptors in floating-point representation consumes significant amounts of memory and the time required to compare descriptors within large datasets becomes longer. Another factor is the proliferation of camera-enabled mobile devices (e.g. phones and tablets) that have limited computational power and storage space. This further necessitates features that compute quickly and are compact in their representation.

This new scale of processing has driven several recent works that propose binary feature detectors and descriptors, promising both increased performance as well as compact representation [6,7,8]. Therefore, a comparative analysis of these new, state-of-the-art techniques is required. At the same time, the analysis must embed itself into the large body of existing analyses to allow comparison.

In this paper, we provide such an analysis. We rely on established evaluation metrics and develop a new taxonomy of all features. To evaluate traditional and

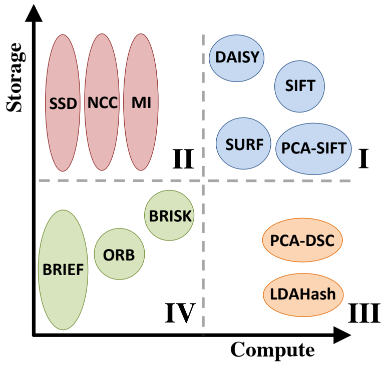

**Fig. 1.** A taxonomy of descriptors based on their computational and storage requirements: I: Real Value Parameterization [14,19,20,21], II: Patch-Based [17], III: Binarized [22,23], and IV: Binary [6,7,8].

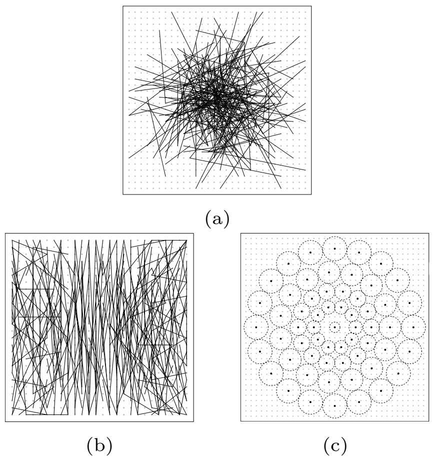

**Fig. 2.** Example patterns of the (a) BRIEF, (b) ORB, and (c) BRISK descriptors

binary features, we propose a comprehensive set of metrics that also overcomes limitations in the previously performed evaluations. Additionally, when selecting datasets, we chose standard existing image sequences, but supplemented them with our own custom datasets to test for missing aspects. The new complete benchmarking scenes cover a wide variety of challenges including changes in scale, rotation, illumination, viewpoint, image quality, and occlusion due to the change in viewpoint of non-planar geometry. We also decouple the detection and description phases, and evaluate the pairwise combinations of different detectors and descriptors. As a result of our analysis, we provide practical guidelines for the strengths and weaknesses of binary descriptors. Finally, we provide an evaluation website for the automatic benchmarking of novel features (using the results of new detectors and descriptors on the datasets used in our analysis).[1]

## 2 Related Work

Feature performance (detection, description, and matching) is important to many computer vision applications. In 2005, Mikolajczyk et al. [9] evaluated affine region detectors, and looked to define the repeatability and accuracy of several affine covariant region detectors. They also provided a set of benchmark image sequences (the Oxford dataset) to test the effects of blur, compression, exposure, scale/rotation, and perspective change, which we leverage for compatibility. Also in 2005, Mikolajczyk and Schmid performed an evaluation of local descriptors [10], comparing complex description techniques and several region detectors. It defined two important metrics, *recall* and *1 − precision*. Later,

[1] `www.cs.unc.edu/feature-evaluation`

Moreels and Perona [11] evaluated several popular detectors and descriptors by analyzing their performance on 3D objects. Strecha et al. [12] published a dense 3D dataset, which we use in our evaluation as it provides 3D LIDAR-based geometry and camera poses. Finally, Aanæs et al. [13] evaluated detectors using a large dataset of known camera positions, controlled illumination, and 3D models.

While many evaluations address the performance of a feature, the recent surge in large-scale feature-based applications draws attention to their runtime requirements. We isolate a feature's computation expense (detection, description, or matching) and the amount of memory required (to store and use) to establish a taxonomy for feature classification (see Figure 1).

Small-scale applications can often afford large computational and memory requirements (corresponding to real value parameterization). Techniques in this category rely on a parameterization of an image region, where each dimension is a floating-point type (or a discretization of a float, excluding binary). These techniques, examined in [10], use image gradients, spatial frequencies, etc. to describe the local image patch and to test for similarity by using the $L^2$ norm, Mahalanobis distance, etc. These descriptors have proven to be effective, and tackle issues such as scale, rotation, viewpoint, or illumination variation. The popular SIFT [14] is in this class. However, increased complexity and robustness comes with an increase in computation and storage requirements. High performance, parallel hardware (e.g. graphics processors) can be used to mitigate higher computational expenses, as shown in [15,16], but even then, descriptor computation can still be the most time-consuming aspect of a system [2].

Therefore, to reduce the computational bottleneck of a system, we address patch-based descriptors. These methods use an image patch surrounding the feature to directly represent it. Distance measures such as sum of squared differences (SSD), normalized cross correlation (NCC), or mutual information (MI) are used to compute pair similarity [17]. The pixels in the patch must be stored, with quadratically increasing requirements for larger patches. However, in large-scale databases (for recognition or reconstruction [18]) the biggest constraints are the matching speed, bandwidth, and the storage required by the descriptor.

The next region in our taxonomy, binarized descriptors, consists of techniques that have high computational but low storage requirements. Binarized descriptors rely on hashing techniques to reduce high-dimensional, real-value parameterizations into compact binary codes [22,23,24,25]. While reducing storage constraints, it also speeds up comparison times through the use of the Hamming distance measure. However, computational requirements are still high as the full real-value parameterization must be computed before the hashing can occur.

The final region in our taxonomy is the binary descriptors. These descriptors have a compact binary representation and limited computational requirements, computing the descriptor directly from pixel-level comparisons. This makes them an attractive solution to many modern applications, especially for mobile platforms where both compute and memory resources are limited.

**Table 1.** Overview of the basic properties of the binary descriptors

| Descriptor | Detector | Rotation Invariant | Scale Invariant |
|---|---|---|---|
| BRIEF | Any | No | No |
| ORB | FAST | Yes | No |
| BRISK | AGAST | Yes | Yes |

## 3 Survey of Binary Descriptors

Recent works have focused on providing methods to directly compute binary descriptors from local image patches. We survey these proposed techniques below.

**Common Principles:** In our analysis, we found that all of the recent binary descriptors possess the same following properties:

- the descriptor is built from a set of pairwise intensity comparisons
- each bit in the descriptor is the result of exactly one comparison
- the sampling pattern is fixed (except for possible scale and rotation)
- Hamming distance is used as a similarity measure

While the binary descriptors use these basic principles, each adds its own unique properties to achieve its design goals. We detail the differences between the binary descriptors (Table 1 highlights their key properties).

**BRIEF (Binary Robust Independent Elementary Features)** are proposed by Calonder et al. [6], and are the simplest of the methods. It uses a sampling pattern consisting of 128, 256, or 512 comparisons (equating to 128, 256, or 512 bits), with sample points selected randomly from an isotropic Gaussian distribution centered at the feature location (see Figure 2(a). Calonder et al. [6] use the SURF detector negating their computational gain, but BRIEF can be coupled with any other detector. Calonder et al. suggest to use BRIEF with the efficient CenSurE detector [26]. Given its simple construction and compact storage, BRIEF has the lowest compute and storage requirements.

**ORB (Oriented FAST and Rotated BRIEF)** was proposed by Rublee et al. [7], and overcomes the lack of rotation invariance of BRIEF. ORB computes a local orientation through the use of an intensity centroid [27], which is a weighted averaging of pixel intensities in the local patch assumed not be coincident with the center of the feature. The orientation is the vector between the feature location and the centroid. While this may seem unstable, it is competitive with the single orientation assignment employed in SIFT [28].

The sampling pattern employed in ORB uses 256 pairwise intensity comparisons, but in contrast to BRIEF, is constructed via machine learning, maximizing the descriptor's variance and minimizing the correlation under various orientation changes (see Figure 2(b) for an example).

When using this descriptor, ORB proposes to use the FAST corner detector [29], noting that FAST does not provide a good measure of cornerness and lacks robustness to multi-scale features. In order to combat this, the Harris [30] corner measure is applied at each keypoint location to provide non-maximal suppression

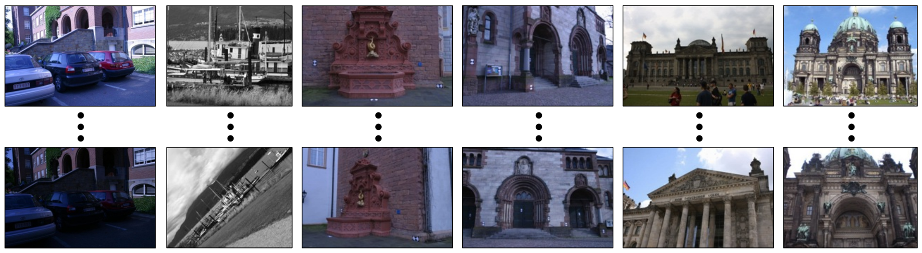

**Fig. 3.** Example images from the evaluation datasets. From left to right, the datasets are Leuven, Boat, fountain-P11, Herz-Jesu-P8, Reichstag, and Berliner-Dom.

within the image, and a limited image scale pyramid is used to detect keypoints of varying sizes. However, no non-maxima suppression is used between the scales, resulting in potential duplicate detections within different pyramid levels.

**BRISK (Binary Robust Invariant Scalable Keypoints)** was proposed by Leutenegger et al. [8] and provides both scale and rotation invariance. In order to compute the feature locations, it uses the AGAST corner detector [31], which improves FAST by increasing speed while maintaining the same detection performance. For scale invariance, BRISK detects keypoints in a scale-space pyramid, performing non-maxima suppression and interpolation across all scales.

To describe the features, the authors turn away from the random or learned patterns of BRIEF and ORB, and instead use a symmetric pattern. Sample points are positioned in concentric circles surrounding the feature, with each sample point representing a Gaussian blurring of its surrounding pixels. The standard deviation of this blurring is increased with the distance from the center of the feature (see Figure 2(c) for an illustration). This may seem similar to the DAISY descriptor [20], but the authors point out that DAISY was designed specifically for dense matching, and captures more information than is needed for keypoint description.

To determine orientation, several long-distance sample point comparisons (e.g. on opposite sides of the descriptor pattern) are used. For each long-distance comparison, the vector displacement between the sample points is stored and weighted by the relative difference in intensity. Then, these weighted vectors are averaged to determine the dominant gradient direction of the patch. The sampling pattern is then scaled and rotated, and the descriptor is built up of 512 short-distance sample point comparisons (e.g. a sample point and its closest neighbors) representing the local gradients and shape within the patch.

Overall, BRISK requires significantly more computation and slightly more storage space than either BRIEF or ORB. This places it in the higher compute, higher storage region of the binary descriptor category of our taxonomy.

## 4 Evaluation

We analyze the performance characteristics of the three recent binary descriptors (BRIEF, ORB, and BRISK), while using state-of-the-art full parametrization descriptors (SIFT and SURF [19]) as a baseline. Besides traditional descriptor performance metrics, we also evaluate the correlation between the detector and descriptor with respect to matching performance. This has received some attention in the past [32], but is important to investigate given its significant practical implications in descriptor post-processing methods (for instance, RANSAC-based estimations [33]). One of the contributions of our comparative analysis is that we test the performance of the studied descriptors with a diverse set of keypoint detectors (Harris, MSER, FAST, ORB, BRISK, SURF, and SIFT).

**Datasets:** To ensure our work is compatible with existing analyses, we use existing datasets to evaluate the performance of the binary descriptors under various transformations. Specifically, we relied on the Oxford dataset provided and described by Mikolajczyk et al. [9] (for evaluations of the effects of image blur, exposure, JPEG compression, combined scale and rotation, and perspective transformations of planar geometry), and the fountain-P11 and Herz-Jesu-P8 datasets from Strecha et al. [12] (for evaluating the effects of perspective transformations of non-planar geometry). Moreover, we complement the existing datasets with our own to test for pure rotation, pure scaling, illumination changes, and the challenges posed by photo collection datasets such as white balance, auto-exposure, image quality, etc. These various datasets also enable us to isolate the effects of each transformation, or in some cases pairs of transformations.

**Performance Metrics:** Mikolajczyk et al. [9,10] propose to use the metrics of *recall*, *repeatability*, and *1 − precision*. They describe useful characteristics of a feature's performance, and are widely used as standard measures. However, there are some subtleties to a feature's performance that are missed by only using these measures. For instance, they fail to capture information about the spatial distribution of the features, as well as the frequency of candidate matches.

We wanted to not only use a comprehensive set of metrics that allow us to embed our analysis into the existing body of work, but we also aimed at evaluating parameters relevant to algorithms relying on the features. As such, we propose a set a five different metrics: *putative match ratio*, *precision*, *matching score*, *recall*, and *entropy*.

The putative match ratio, $Putative\ Match\ Ratio = \#\,Putative\ Matches\ /\ \#\,Features$, addresses the selectivity of the descriptor and describes what fraction of the detected features will be initially identified as a match (though potentially incorrect). We define a putative match to be a single pairing of keypoints, where a keypoint cannot be matched to more than one other. Keypoints that are outside of the bounds of the second image (once they are transformed based on the true camera positioning) are not counted.

The value of the putative match ratio is directly influenced by the matching criteria. A less restrictive matching criteria will generate a higher putative match ratio, whereas a criteria that is too restrictive will discard potentially valid matches and will decrease the putative match ratio. Another influencing factor is the distinctiveness of the descriptors under consideration. If many descriptors are highly similar (have small distance values between them), this creates confusion in the matching criteria and can drive down the putative match ratio.

The precision, *Precision = # Correct Matches / # Putative Matches* [10]), defines the number of correct matches out of the set of putative matches (the inlier ratio). In this equation, the number of correct matches are those putative matches that are geometrically verified based on the known camera positions. The ratio has significant performance consequences for robust estimation modules that use feature matches, such as RANSAC [33], where execution times increase exponentially as the inlier ratio decreases. It is also influenced by many of the same factors that influenced the putative match ratio, but the consequences are different. For instance, while a less restrictive matching criteria will increase the putative match ratio, it will decrease the precision as a higher number of incorrect matches will be generated. Additionally, highly similar descriptors, which drove down the putative match ratio, will also decrease the precision, as confusion in the matching step will also generate a higher number of incorrect matches.

The matching score, *Matching Score = # Correct Matches / # Features* [10] is equivalent to the multiplication of the putative match ratio and precision. It describes the number of initial features that will result in correct matches, and like the previous two metrics, the matching score can be influenced by indistinct descriptors and the matching criteria. Overall, the matching score describes how well the descriptor is performing and is influenced by the descriptor's robustness to transformations present in the data.

Recall, *Recall = # Correct Matches / # Correspondences* (defined in [10]), quantifies how many of the possible correct matches were actually found. The correspondences are the matches that should have been identified given the keypoint locations in both images. While this value is dependent on the detector's ability to generate correspondences, recall shares the same influences as the matching score. For instance, a low recall could mean that the descriptors are indistinct, the matching criterion is too strict, or the data is too complex.

The final metric, entropy (used by Zitnick and Ramnath [34]) addresses the influence of the detector on a descriptor. The purpose of this metric is to compute the amount of spread or randomness in the spatial distribution of the keypoints in the image. This is important as too little spread increases the possibility of confusion in the descriptor matching phase due to keypoint clusters.

To compute the entropy, we create a 2D evenly-spaced binning of the feature points. Each point's contribution to a given bin is weighted by a Gaussian relative to its distance to the bin's center. A bin $b(p)$ at position $p = (x, y)$ equals $b(p) = \frac{1}{Z}\sum_{m \in M} G(\|p - m\|)$ where $m$ is a keypoint in the full set $M$ of detected keypoints, and $G$ is a Gaussian. A constant of $1/Z$ is added so that the sum of all bins evaluates to 1. This binning allows us to compute the entropy: $\mathit{Entropy} = \sum_p -b(p) \log b(p)$. Even though entropy is dataset dependent, the relative value of the entropy is useful in identifying detector spatial distribution behaviors such as non-random clustering of keypoint detections (as seen in Figure 4).

We can now quantify how many features were reported as matching (putative match ratio), how many actually matched (precision and matching score), how many were matched out of those possible (recall), and the spread of the detector's keypoints (entropy). Next, we will describe our evaluation framework.

**Test Setup:** For each dataset, we detect and describe features in each of its images, and use the features in the first image as a reference when matching to all further images in the set. In order to test the detector/descriptor pairings, we had to address several subtleties. For instance, there was a mismatch when a scale invariant descriptor was combined with a detector that was not scale invariant, and vice versa. Additionally, combining detectors and descriptors that were both scale invariant was not trivial, as they typically each use their own method of defining what a feature scale is. In both cases, we simply discarded the scale information, and computed the descriptor at the native image resolution.

In addition, we addressed mismatches in rotation invariance between various detectors and descriptors. We solved this by isolating the orientation computation to the description phase, so that the descriptor overrides any orientation provided by the detector. We next discuss the matching criteria that we use.

**Match Criteria:** To compute putative matches, we adopt a ratio style test that has proven to be effective [11], [14]. This test compares the ratio of distances between the two best matches for a given keypoint, and rejects the match if the ratio is above a threshold of 0.8 for all tests (the same used in [14]).

To determine the correctness of a match, we use ground truth data to warp keypoints from the first image of the dataset into all remaining images. The warping is achieved either by homographies (provided in the Oxford dataset [9] as well as our non-photo-collection supplemental datasets) or by using ground truth 3D geometry (provided in the Strecha dataset [12]) to project the points into the known cameras. In both cases, match points that are within 2.5 pixels of each other are assumed to be correct. This threshold was chosen empirically as it provided a good balance between pixel-level corner detection (such as Harris [30] and FAST [29]), and the centers of blobs in blob-style detectors (SIFT [14] and SURF [19]). For the photo-collection datasets, we use known camera positions to project points from the first image as epipolar lines in the other images, and once again apply a 2.5 pixel distance threshold test. While we could have used the three camera arrangement test described by Moreels and Perona [11], we opted for the single epipolar line approach as it allowed us to compare only a given pair of images, without the need for a third image to provide correspondences and closely mimics typical uses of features.

## 5 Analysis and Results

In our analysis we tested the individual effects of various geometric and photometric transformations, as well as several of their combinations to gain a better and more complete understanding of the detectors' and descriptors' performance. Figure 5 provides the results for all tested dataset categories. The individual values making up each measurement (# putative matches, features, etc.) were

**Table 2.** This table provides statistics for the detectors and descriptors used in our evaluation. For storage, we used a 32 byte BRIEF, and the values in parenthesis for SURF and SIFT are the number of required bytes if the descriptors are stored as floats. For the timings, values in parenthesis are GPU implementations ([15], [16], or our own).

| Detector/Descriptor | BRIEF | ORB | BRISK | SURF | SIFT | Harris | MSER | FAST |
|---|---|---|---|---|---|---|---|---|
| Avg # Features | n/a | 13427 | 7771 | 3766 | 4788 | 2543 | 693 | 8166 |
| Detector Entropy | n/a | 12.10 | 12.33 | 12.26 | 12.34 | 11.84 | 10.74 | 12.52 |
| Detector ms/image | n/a | 17 | 43 | 377(19) | 572(25) | 78(4.7) | 117 | 2.7 |
| Descriptor $\mu$s/feature | 4.4(0.4) | 4.8 | 12.9 | 143(6.6) | 314(19) | n/a | n/a | n/a |
| Storage bytes/feature | 16,32,64 | 32 | 64 | 64(256) | 128(512) | n/a | n/a | n/a |

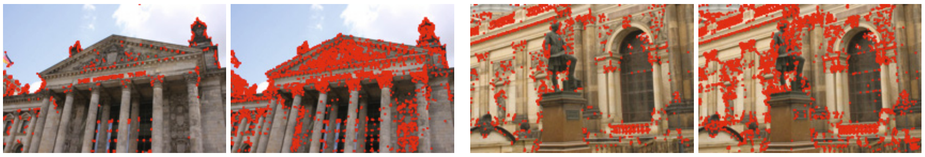
**Fig. 4.** This figure shows the distribution of keypoints (for two datasets) from the Harris (left image in the pair) and FAST (right image in the pair) corner detectors. The FAST detector not only detects more keypoints, but has a higher entropy for its detections.

first summed across each pairwise comparison, and one final average was computed.

**Detector Performance:** As mentioned before, entropy can be used as a measure of the randomness of the keypoint locations, penalizing detectors that spatially cluster their keypoints. In order to compute the entropy across all of the datasets, we perform a weighted average of the individual entropies to account for the different contributions of the datasets (results are in Table 2).

We see several notable attributes. First, FAST has the highest entropy, which can be attributed to its good spread of points and the shear number of detections that occurred. A higher number of keypoints does not necessarily correspond to increased entropy, but it can help if more spatial bins in the entropy equation are populated. The second group of detectors, SIFT, BRISK, and SURF, have the next highest entropies, which is expected as Zitnick's and Ramnath's evaluation [34] noted that a blob-style detector (difference of Gaussian) have high entropies.

On the other end, Harris and MSER reported the lowest two entropy values. The primary reason for this is a lower average number of keypoint detections (especially in the case of MSER), which leads to less spatial bins being populated. While these detectors are still very viable, it is important to note their lower detection rates and potential differences in keypoint distribution (one such example is provided in Figure 4).

**Descriptor Performance:** One of the most compelling motivations for the use of binary descriptors is their efficiency and compactness. Table 2 shows the timing results for the various detectors and descriptors that we used in our system. The implementations for ORB and BRISK were obtained from the authors, while all others came from OpenCV 2.3 [35]. The code was run on a computer with an Intel Xeon 2.67GHz processor, 12GB of RAM, NVIDIA GTX 285, and Microsoft Windows 7. From the results, we can see that the binary descriptors (and many of their paired detectors) are an order of magnitude faster than the SURF or SIFT alternatives. For mobile devices, while the overall timings would change, a speedup would still be realized as the binary descriptors are algorithmically more efficient. Table 2 also lists the storage requirements for the descriptors. In some cases, the binary descriptors can be constructed such that they are on par with efficient SURF representations, but overall, binary descriptors reduce storage to a half or to a quarter. We are assuming that the real value parameterization descriptors are stored in a quantized form (1 byte per dimension). If instead they are stored as floating-point values, the storage savings of binary features are even more significant.

Figure 5 shows the results for our evaluation of detector and descriptor pairings. Upon a close inspection, several interesting conclusions become apparent.

**Non-geometric Transforms:** Non-geometric transforms consist of those that are image-capture dependent, and do not rely on the viewpoint (e.g. blur, JPEG compression, exposure, and illumination). Challenges in these datasets involve less-distinct image gradients and changes to the relative difference between pixels due to compression or lighting change.

Upon inspecting the results, BRIEF's performance is very favorable. It outperforms ORB and BRISK in many of the categories (even when compared to SURF or SIFT), though the precision is noticeably lower than ORB or BRISK. The key insight to BRIEF's good performance is that it has a fixed pattern (no scale or rotation invariance), and is designed to accurately represent the underlying gradients of an image patch (which are stable for monotonic color transformations). In regard to the lower precision, as we mentioned before, the precision is decreased when the matching criteria are either less restrictive, or the features are not distinct enough. We enforced the same matching criteria for each descriptor, hence performance differences are due to the lack of distinctiveness. The pattern used by ORB is specifically trained to maximize distinctiveness, and BRISK's descriptor is much more complex (larger number of samples with each sample being a comparison of blurred pixels), which allows it to be much more certain of the matches it generates. Therefore, it is not surprising that BRIEF's precision would suffer slightly compared to ORB or BRISK.

For the choice of detector, both SURF and Harris perform well under these non-geometric transformations. The key is that both are very repeatable in these images. SURF (being a blob detector) does well for the blur dataset, as the smoothing of the gradients negatively impacts corner detectors like Harris. SIFT (also being a blob detector) does not do as well as the accuracy of its keypoint detections decreases as the scale of the blob increases. This is an artifact of the downsampling in the image pyramid, where SURF overcomes this by applying larger filters at the native resolution. For the non-blurred datasets, Harris does well because high gradient change still exists at corners under color changes.

**Affine Image Transforms:** The affine image transforms that we used in our testing consist of image plane rotation and scaling. Overall, SIFT was the best, but ORB and BRISK still performed well. As is expected, BRIEF performs poorly under large rotation or scale change compared to the other descriptors (as it is not designed to handle those effects). For pure scale change, ORB is better than BRIEF although there is no rotation change in the image. This can be attributed to the limited scale space used by the ORB detector, allowing it to detect FAST corners at several scales. However, of all the binary descriptors, BRISK performs the best, as it is scale invariant.

For pure rotations, the ORB detector/descriptor combination performs better than the BRISK detector/descriptor. However, the FAST detector paired with BRISK is a top performer (as well as the Harris detector paired with ORB), competing on the same level as SIFT. The insight to this result is that both FAST and Harris are not scale invariant, and excel because there is no scale change in the images. The better performances of BRISK and ORB when paired with other detectors are unexpected, but only help to highlight the importance of considering various combinations when deciding on a detector and descriptor.

Finally, when analyzing combined scale and rotation changes, BRISK takes the lead of the binary descriptors. However, SIFT and SURF have competitive matching scores, but significantly higher recalls. This difference in matching score and recall speaks to the performance of the detectors. They detected almost the same percentage of correct matches (the matching score), SURF and SIFT's number of correspondences must have been lower in order for their recall to be higher. This means that for basic scale and rotation combinations, BRISK's detector is more repeatable than SURF or SIFT's.

Turning our attention back to BRIEF, one way to overcome BRIEF's sensitivity to rotation when computing on mobile devices is via inertial sensor-aligned features [36] which aligns a descriptor with the current gravity direction of the mobile device (assuming that the device has an accelerometer), allowing for repeatable detections of the same feature as the device changes orientation.

**Perspective Transforms** are the result of changes in viewpoint. The biggest issues faced in these sets of images are occlusions due to depth edges, as well as perspective warping. For binary descriptors, BRIEF surprisingly has a slight lead in recall and matching score over ORB and BRISK, given its limited complexity. However, in most of the perspective evaluation datasets, there is no significant change in scale, and the orientation of the images are the same. This is not the case in the photo-collection datasets, which include a higher variety of feature scales. Even then, BRIEF still takes the lead over the other binary descriptors. So, even though there is considerable viewpoint change throughout all of the perspective datasets, the upright nature of all of the images allows the fixed orientation of BRIEF to excel.

**Putative Match Ratio**

| | BF | OB | BK | SF | ST |
|---|---|---|---|---|---|
| H | 25 | 11 | 11 | | |
| MR | 19 | 10 | 9 | | |
| FT | 23 | 10 | 11 | | |
| OB | 13 | 10 | 8 | | |
| BK | 20 | 9 | 9 | | |
| SF | 31 | 15 | 12 | 15 | |
| ST | 19 | 8 | 9 | | 10 |

**Precision**

| | BF | OB | BK | SF | ST |
|---|---|---|---|---|---|
| H | 67 | 78 | 84 | | |
| MR | 65 | 71 | 69 | | |
| FT | 68 | 72 | 85 | | |
| OB | 70 | 78 | 87 | | |
| BK | 67 | 77 | 73 | | |
| SF | 72 | 82 | 79 | 71 | |
| ST | 63 | 76 | 76 | | 73 |

**Matching Score**

| | BF | OB | BK | SF | ST |
|---|---|---|---|---|---|
| H | 17 | 8 | 9 | | |
| MR | 12 | 7 | 6 | | |
| FT | 15 | 7 | 9 | | |
| OB | 9 | 8 | 7 | | |
| BK | 13 | 7 | 6 | | |
| SF | 22 | 12 | 9 | 11 | |
| ST | 12 | 6 | 7 | | 7 |

**Recall**

| | BF | OB | BK | SF | ST |
|---|---|---|---|---|---|
| H | 44 | 22 | 24 | | |
| MR | 40 | 22 | 19 | | |
| FT | 39 | 10 | 24 | | |
| OB | 19 | 16 | 12 | | |
| BK | 31 | 9 | 14 | | |
| SF | 55 | 29 | 22 | 28 | |
| ST | 38 | 18 | 18 | | 24 |

(a)

**Putative Match Ratio**

| | BF | OB | BK | SF | ST |
|---|---|---|---|---|---|
| H | 68 | 54 | 51 | | |
| MR | 43 | 31 | 29 | | |
| FT | 50 | 33 | 35 | | |
| OB | 35 | 36 | 30 | | |
| BK | 50 | 35 | 33 | | |
| SF | 59 | 43 | 30 | 47 | |
| ST | 35 | 23 | 28 | | 29 |

**Precision**

| | BF | OB | BK | SF | ST |
|---|---|---|---|---|---|
| H | 95 | 97 | 97 | | |
| MR | 83 | 88 | 86 | | |
| FT | 91 | 96 | 97 | | |
| OB | 95 | 97 | 98 | | |
| BK | 93 | 97 | 94 | | |
| SF | 92 | 96 | 94 | 90 | |
| ST | 85 | 93 | 94 | | 91 |

**Matching Score**

| | BF | OB | BK | SF | ST |
|---|---|---|---|---|---|
| H | 64 | 53 | 49 | | |
| MR | 36 | 27 | 25 | | |
| FT | 46 | 32 | 34 | | |
| OB | 34 | 35 | 30 | | |
| BK | 46 | 34 | 32 | | |
| SF | 54 | 41 | 28 | 42 | |
| ST | 30 | 22 | 27 | | 26 |

**Recall**

| | BF | OB | BK | SF | ST |
|---|---|---|---|---|---|
| H | 80 | 64 | 63 | | |
| MR | 53 | 40 | 33 | | |
| FT | 69 | 47 | 54 | | |
| OB | 47 | 49 | 35 | | |
| BK | 59 | 43 | 41 | | |
| SF | 74 | 56 | 39 | 61 | |
| ST | 55 | 40 | 40 | | 55 |

(b)

**Putative Match Ratio**

| | BF | OB | BK | SF | ST |
|---|---|---|---|---|---|
| H | 39 | 28 | 26 | | |
| MR | 45 | 31 | 30 | | |
| FT | 53 | 38 | 39 | | |
| OB | 36 | 36 | 35 | | |
| BK | 44 | 32 | 28 | | |
| SF | 43 | 31 | 25 | 30 | |
| ST | 34 | 25 | 29 | | 34 |

**Precision**

| | BF | OB | BK | SF | ST |
|---|---|---|---|---|---|
| H | 80 | 91 | 93 | | |
| MR | 76 | 76 | 72 | | |
| FT | 84 | 93 | 95 | | |
| OB | 89 | 93 | 96 | | |
| BK | 81 | 90 | 87 | | |
| SF | 80 | 88 | 82 | 78 | |
| ST | 76 | 88 | 87 | | 86 |

**Matching Score**

| | BF | OB | BK | SF | ST |
|---|---|---|---|---|---|
| H | 31 | 26 | 24 | | |
| MR | 34 | 24 | 22 | | |
| FT | 44 | 36 | 37 | | |
| OB | 32 | 34 | 34 | | |
| BK | 36 | 29 | 24 | | |
| SF | 34 | 28 | 21 | 23 | |
| ST | 26 | 22 | 25 | | 29 |

**Recall**

| | BF | OB | BK | SF | ST |
|---|---|---|---|---|---|
| H | 63 | 51 | 52 | | |
| MR | 63 | 44 | 39 | | |
| FT | 65 | 51 | 58 | | |
| OB | 51 | 53 | 45 | | |
| BK | 55 | 43 | 37 | | |
| SF | 73 | 58 | 44 | 53 | |
| ST | 58 | 48 | 48 | | 74 |

(c)

**Putative Match Ratio**

| | BF | OB | BK | SF | ST |
|---|---|---|---|---|---|
| H | 13 | 8 | 7 | | |
| MR | 15 | 10 | 9 | | |
| FT | 11 | 7 | 7 | | |
| OB | 6 | 5 | 5 | | |
| BK | 9 | 6 | 6 | | |
| SF | 10 | 5 | 4 | 8 | |
| ST | 7 | 5 | 6 | | 10 |

**Precision**

| | BF | OB | BK | SF | ST |
|---|---|---|---|---|---|
| H | 64 | 76 | 82 | | |
| MR | 52 | 41 | 36 | | |
| FT | 71 | 76 | 84 | | |
| OB | 78 | 80 | 89 | | |
| BK | 69 | 73 | 79 | | |
| SF | 57 | 60 | 57 | 37 | |
| ST | 49 | 54 | 61 | | 55 |

**Matching Score**

| | BF | OB | BK | SF | ST |
|---|---|---|---|---|---|
| H | 8 | 6 | 5 | | |
| MR | 8 | 4 | 3 | | |
| FT | 7 | 5 | 6 | | |
| OB | 4 | 4 | 4 | | |
| BK | 6 | 4 | 4 | | |
| SF | 6 | 3 | 2 | 3 | |
| ST | 4 | 3 | 3 | | 6 |

**Recall**

| | BF | OB | BK | SF | ST |
|---|---|---|---|---|---|
| H | 32 | 23 | 22 | | |
| MR | 47 | 24 | 18 | | |
| FT | 25 | 17 | 21 | | |
| OB | 13 | 13 | 11 | | |
| BK | 20 | 14 | 15 | | |
| SF | 35 | 21 | 15 | 19 | |
| ST | 24 | 17 | 18 | | 40 |

(d)

**Putative Match Ratio**

| | BF | OB | BK | SF | ST |
|---|---|---|---|---|---|
| H | 19 | 10 | 12 | | |
| MR | 10 | 8 | 9 | | |
| FT | 15 | 7 | 12 | | |
| OB | 5 | 21 | 5 | | |
| BK | 11 | 7 | 40 | | |
| SF | 15 | 8 | 8 | 56 | |
| ST | 14 | 7 | 10 | | 57 |

**Precision**

| | BF | OB | BK | SF | ST |
|---|---|---|---|---|---|
| H | 80 | 80 | 87 | | |
| MR | 69 | 48 | 53 | | |
| FT | 77 | 73 | 89 | | |
| OB | 79 | 92 | 91 | | |
| BK | 74 | 73 | 78 | | |
| SF | 75 | 67 | 70 | 81 | |
| ST | 76 | 72 | 82 | | 96 |

**Matching Score**

| | BF | OB | BK | SF | ST |
|---|---|---|---|---|---|
| H | 15 | 8 | 11 | | |
| MR | 7 | 4 | 5 | | |
| FT | 11 | 6 | 10 | | |
| OB | 4 | 19 | 5 | | |
| BK | 8 | 5 | 31 | | |
| SF | 11 | 5 | 6 | 45 | |
| ST | 11 | 5 | 8 | | 54 |

**Recall**

| | BF | OB | BK | SF | ST |
|---|---|---|---|---|---|
| H | 24 | 13 | 17 | | |
| MR | 10 | 6 | 7 | | |
| FT | 19 | 9 | 17 | | |
| OB | 6 | 29 | 6 | | |
| BK | 11 | 7 | 42 | | |
| SF | 17 | 8 | 8 | 69 | |
| ST | 16 | 7 | 10 | | 80 |

(e)

**Putative Match Ratio**

| | BF | OB | BK | SF | ST |
|---|---|---|---|---|---|
| H | 3 | 69 | 64 | | |
| MR | 5 | 44 | 42 | | |
| FT | 3 | 62 | 68 | | |
| OB | 1 | 62 | 58 | | |
| BK | 2 | 52 | 51 | | |
| SF | 4 | 44 | 38 | 43 | |
| ST | 3 | 41 | 50 | | 58 |

**Precision**

| | BF | OB | BK | SF | ST |
|---|---|---|---|---|---|
| H | 14 | 93 | 88 | | |
| MR | 18 | 90 | 78 | | |
| FT | 15 | 91 | 93 | | |
| OB | 12 | 86 | 94 | | |
| BK | 12 | 89 | 86 | | |
| SF | 13 | 88 | 79 | 82 | |
| ST | 13 | 93 | 84 | | 95 |

**Matching Score**

| | BF | OB | BK | SF | ST |
|---|---|---|---|---|---|
| H | 0 | 64 | 56 | | |
| MR | 1 | 40 | 33 | | |
| FT | 0 | 57 | 64 | | |
| OB | 0 | 53 | 55 | | |
| BK | 0 | 46 | 44 | | |
| SF | 0 | 38 | 30 | 35 | |
| ST | 0 | 38 | 42 | | 55 |

**Recall**

| | BF | OB | BK | SF | ST |
|---|---|---|---|---|---|
| H | 0 | 68 | 62 | | |
| MR | 1 | 50 | 39 | | |
| FT | 0 | 63 | 74 | | |
| OB | 0 | 62 | 53 | | |
| BK | 0 | 52 | 48 | | |
| SF | 1 | 57 | 45 | 55 | |
| ST | 1 | 50 | 48 | | 79 |

(f)

**Putative Match Ratio**

| | BF | OB | BK | SF | ST |
|---|---|---|---|---|---|
| H | 4 | 6 | 6 | | |
| MR | 6 | 6 | 5 | | |
| FT | 2 | 4 | 6 | | |
| OB | 1 | 8 | 4 | | |
| BK | 2 | 4 | 13 | | |
| SF | 3 | 4 | 4 | 14 | |
| ST | 3 | 4 | 4 | | 13 |

**Precision**

| | BF | OB | BK | SF | ST |
|---|---|---|---|---|---|
| H | 34 | 65 | 81 | | |
| MR | 19 | 40 | 35 | | |
| FT | 23 | 64 | 83 | | |
| OB | 20 | 83 | 86 | | |
| BK | 27 | 68 | 84 | | |
| SF | 20 | 48 | 58 | 68 | |
| ST | 18 | 55 | 72 | | 84 |

**Matching Score**

| | BF | OB | BK | SF | ST |
|---|---|---|---|---|---|
| H | 1 | 4 | 5 | | |
| MR | 1 | 2 | 2 | | |
| FT | 1 | 3 | 5 | | |
| OB | 0 | 7 | 3 | | |
| BK | 0 | 3 | 11 | | |
| SF | 1 | 2 | 2 | 9 | |
| ST | 0 | 2 | 3 | | 11 |

**Recall**

| | BF | OB | BK | SF | ST |
|---|---|---|---|---|---|
| H | 2 | 6 | 9 | | |
| MR | 5 | 9 | 8 | | |
| FT | 1 | 4 | 7 | | |
| OB | 0 | 8 | 3 | | |
| BK | 1 | 3 | 14 | | |
| SF | 1 | 5 | 6 | 25 | |
| ST | 1 | 5 | 7 | | 30 |

(g)

**Putative Match Ratio**

| | BF | OB | BK | SF | ST |
|---|---|---|---|---|---|
| H | 29 | 14 | 17 | | |
| MR | 15 | 9 | 9 | | |
| FT | 25 | 13 | 17 | | |
| OB | 12 | 15 | 13 | | |
| BK | 19 | 11 | 13 | | |
| SF | 18 | 10 | 10 | 15 | |
| ST | 18 | 9 | 13 | | 24 |

**Precision**

| | BF | OB | BK | SF | ST |
|---|---|---|---|---|---|
| H | 88 | 87 | 90 | | |
| MR | 76 | 68 | 70 | | |
| FT | 84 | 85 | 89 | | |
| OB | 89 | 88 | 92 | | |
| BK | 86 | 83 | 87 | | |
| SF | 78 | 73 | 75 | 73 | |
| ST | 83 | 80 | 81 | | 86 |

**Matching Score**

| | BF | OB | BK | SF | ST |
|---|---|---|---|---|---|
| H | 26 | 13 | 15 | | |
| MR | 11 | 6 | 6 | | |
| FT | 21 | 11 | 15 | | |
| OB | 11 | 13 | 12 | | |
| BK | 16 | 9 | 11 | | |
| SF | 14 | 7 | 7 | 11 | |
| ST | 15 | 7 | 11 | | 20 |

**Recall**

| | BF | OB | BK | SF | ST |
|---|---|---|---|---|---|
| H | 36 | 18 | 22 | | |
| MR | 25 | 13 | 14 | | |
| FT | 28 | 8 | 21 | | |
| OB | 13 | 15 | 12 | | |
| BK | 22 | 7 | 15 | | |
| SF | 32 | 17 | 17 | 25 | |
| ST | 28 | 14 | 17 | | 42 |

(h)

**Putative Match Ratio**

| | BF | OB | BK | SF | ST |
|---|---|---|---|---|---|
| H | 22 | 13 | 14 | | |
| MR | 17 | 11 | 11 | | |
| FT | 21 | 12 | 15 | | |
| OB | 9 | 13 | 9 | | |
| BK | 15 | 9 | 15 | | |
| SF | 16 | 9 | 9 | 18 | |
| ST | 15 | 9 | 13 | | 22 |

**Precision**

| | BF | OB | BK | SF | ST |
|---|---|---|---|---|---|
| H | 60 | 66 | 73 | | |
| MR | 55 | 39 | 37 | | |
| FT | 73 | 73 | 81 | | |
| OB | 77 | 80 | 85 | | |
| BK | 63 | 62 | 72 | | |
| SF | 58 | 54 | 58 | 61 | |
| ST | 69 | 68 | 75 | | 84 |

**Matching Score**

| | BF | OB | BK | SF | ST |
|---|---|---|---|---|---|
| H | 13 | 9 | 10 | | |
| MR | 9 | 4 | 4 | | |
| FT | 15 | 9 | 12 | | |
| OB | 7 | 10 | 8 | | |
| BK | 9 | 5 | 11 | | |
| SF | 10 | 5 | 5 | 11 | |
| ST | 11 | 6 | 10 | | 19 |

**Recall**

| | BF | OB | BK | SF | ST |
|---|---|---|---|---|---|
| H | 30 | 19 | 23 | | |
| MR | 32 | 15 | 13 | | |
| FT | 30 | 17 | 24 | | |
| OB | 12 | 18 | 11 | | |
| BK | 19 | 10 | 20 | | |
| SF | 28 | 14 | 15 | 33 | |
| ST | 24 | 13 | 18 | | 44 |

(i)

**Putative Match Ratio**

| | BF | OB | BK | SF | ST |
|---|---|---|---|---|---|
| H | 11 | 6 | 6 | | |
| MR | 11 | 7 | 7 | | |
| FT | 9 | 5 | 5 | | |
| OB | 4 | 6 | 4 | | |
| BK | 7 | 4 | 7 | | |
| SF | 10 | 6 | 5 | 11 | |
| ST | 8 | 4 | 5 | | 11 |

**Precision**

| | BF | OB | BK | SF | ST |
|---|---|---|---|---|---|
| H | 81 | 78 | 84 | | |
| MR | 76 | 72 | 63 | | |
| FT | 84 | 77 | 86 | | |
| OB | 87 | 84 | 88 | | |
| BK | 83 | 77 | 86 | | |
| SF | 79 | 73 | 75 | 79 | |
| ST | 76 | 71 | 75 | | 83 |

**Matching Score**

| | BF | OB | BK | SF | ST |
|---|---|---|---|---|---|
| H | 9 | 5 | 5 | | |
| MR | 8 | 5 | 4 | | |
| FT | 7 | 4 | 5 | | |
| OB | 4 | 5 | 3 | | |
| BK | 6 | 3 | 6 | | |
| SF | 8 | 4 | 4 | 8 | |
| ST | 6 | 3 | 4 | | 10 |

**Recall**

| | BF | OB | BK | SF | ST |
|---|---|---|---|---|---|
| H | 8 | 5 | 5 | | |
| MR | 8 | 5 | 4 | | |
| FT | 7 | 4 | 5 | | |
| OB | 4 | 5 | 3 | | |
| BK | 6 | 3 | 5 | | |
| SF | 8 | 4 | 4 | 9 | |
| ST | 6 | 3 | 3 | | 10 |

(j)

**Fig. 5.** Results for (a) blur, (b) JPEG compression, (c) exposure, (d) day-to-night illumination, (e) scale, (f) rotation, (g) scale and rotation, (h) perspective with planar scene, (i) perspective with non-planar scene, (j) photo-collection. Rows are detectors, columns are descriptors: H=Harris, MR=MSER, FT=FAST, BF=BRIEF, OB=ORB, BK=BRISK, SF=SURF, ST=SIFT. Values are in percent (blue 0%, red 100%).

Additionally, it is interesting to note that the Harris corner detector does well when coupled with BRIEF. Harris is not scale invariant, but the nature of corners is that there will be high gradient change even under perspective distortion allowing for repeatable detections.

Another observation is that the precision of BRISK is once again higher than its binary competitors. This points to the descriptiveness of BRISK, allowing it to achieve very high quality matches. Once again, however, SIFT provides the best performance, and proves to be the most robust to changes in perspective.

## 6 Conclusion

From the analysis of our results, we highlight several pertinent observations, each of which was driven by the metrics used in our evaluation. Moreover, by proposing and integrating a comprehensive set of measures and datasets, we are able to gain insights into the various factors that influence a feature's behavior.

First, consider BRIEF's performance under non-geometric and perspective transforms. In both cases, BRIEF excelled because the images contained similar scales and orientations. This highlights the importance of leveraging any additional knowledge about the data being processed (e.g. similar scales/orientations or orientation from inertial sensors) as it impacts the performance of the binary descriptors. The key idea is that a binary descriptor will suffer in performance when it takes into account a transform not present in the data.

As a result of our evaluation of detector/descriptor pairings, we observed that the best performance did not always correspond to the original authors' recommendations. This leads us to our second observation that by considering the effect a detector has on a descriptor, we enable the educated assignment of detectors to descriptors depending on the expected properties of the data.

Finally, for all datasets except the non-geometric transforms, SIFT was the best. Although achieving a gain in matching rate, some applications using SIFT may unnecessarily forfeit the significant speed gains made possible by binary descriptors (Table 2). This performance tradeoff makes binary features a viable choice whenever the application provides robustness against additional outliers.

**Acknowledgments.** This work was supported by NSF IIS-0916829 and the Department of Energy under Award DE-FG52-08NA28778.

## References

1. Chang, H.J., et al.: P-SLAM: Simultaneous Localization and Mapping With Environmental-Structure Prediction. IEEE Trans. Robot. 23(2), 281–293 (2007)
2. Frahm, J.-M., Fite-Georgel, P., Gallup, D., Johnson, T., Raguram, R., Wu, C., Jen, Y.-H., Dunn, E., Clipp, B., Lazebnik, S., Pollefeys, M.: Building Rome on a Cloudless Day. In: Daniilidis, K., Maragos, P., Paragios, N., et al. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 368–381. Springer, Heidelberg (2010)

3. Snavely, N., Seitz, S.M., Szeliski, R.: Photo Tourism: Exploring Photo Collections in 3D. In: SIGGRAPH Conference Proceedings, pp. 835–846 (2006)
4. Nister, D., Stewenius, H.: Scalable Recognition with a Vocabulary Tree. In: CVPR, pp. 2161–2168 (2006)
5. Brown, M., Lowe, D.G.: Recognising Panoramas. In: ICCV, pp. 1218–1225 (2003)
6. Calonder, M., Lepetit, V., Strecha, C., Fua, P.: BRIEF: Binary Robust Independent Elementary Features. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 778–792. Springer, Heidelberg (2010)
7. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: ORB: An Efficient Alternative to SIFT or SURF. In: ICCV, pp. 2564–2571 (2011)
8. Leutenegger, S., Chli, M., Siegwart, R.: BRISK: Binary Robust Invariant Scalable Keypoints. In: ICCV, pp. 2548–2555 (2011)
9. Mikolajczyk, K., et al.: A Comparison of Affine Region Detectors. IJCV 65(1-2), 43–72 (2005)
10. Mikolajczyk, K., Schmid, C.: A Performance Evaluation of Local Descriptors. IEEE Trans. PAMI 27(10), 1615–1630 (2005)
11. Moreels, P., Perona, P.: Evaluation of Features Detectors and Descriptors based on 3D Objects. IJCV 73(3), 263–284 (2007)
12. Strecha, C., et al.: On Benchmarking Camera Calibration and Multi-View Stereo for High Resolution Imagery. In: CVPR, pp. 1–8 (2008)
13. Aanæs, H., et al.: Interesting Interest Points. IJCV 97(1), 18–35 (2012)
14. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV 60(2), 91–110 (2004)
15. Wu, C.: SiftGPU (2007), `http://cs.unc.edu/~ccwu/siftgpu`
16. Schulz, A., et al.: CUDA SURF - A real-time implementation for SURF (2010), `http://www.d2.mpi-inf.mpg.de/surf`
17. Hirschmüller, H., Scharstein, D.: Evaluation of Cost Functions for Stereo Matching. In: CVPR, pp. 1–8 (2007)
18. Agarwal, S., et al.: Building Rome in a Day. In: ICCV, pp. 72–79 (2009)
19. Bay, H., et al.: Speeded-Up Robust Features (SURF). Comp. Vis. and Image Understanding 110(3), 346–359 (2008)
20. Tola, E., Lepetit, V., Fua, P.: DAISY: An Efficient Dense Descriptor Applied to Wide Baseline Stereo. IEEE Trans. PAMI 32(5), 815–830 (2010)
21. Ke, Y., Sukthankar, R.: PCA-SIFT: a more distinctive representation for local image descriptors. In: CVPR, pp. 506–513 (2004)
22. Strecha, C., et al.: LDAHash: Improved Matching with Smaller Descriptors. IEEE Trans. PAMI 34(1), 66–78 (2012)
23. Yeo, C., Ahammad, P., Ramchandran, K.: Coding of Image Feature Descriptors for Distributed Rate-efficient Visual Correspondences. IJCV 94(3), 267–281 (2011)
24. Raginsky, M., Lazebnik, S.: Locality-Sensitive Binary Codes from Shift-Invariant Kernels. In: Advances in Neural Info. Processing Systems, pp. 1509–1517 (2009)
25. Gong, Y., Lazebnik, S.: Iterative Quantization: A Procrustean Approach to Learning Binary Codes. In: CVPR, pp. 817–824 (2011)
26. Agrawal, M., Konolige, K., Blas, M.R.: CenSurE: Center Surround Extremas for Realtime Feature Detection and Matching. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 102–115. Springer, Heidelberg (2008)
27. Rosin, P.L.: Measuring Corner Properties. Comp. Vis. and Image Understanding, 291–307 (1999)
28. Gauglitz, S., et al.: Improving Keypoint Orientation Assignment. In: BMVC (2011)

29. Rosten, E., Porter, R., Drummond, T.: Faster and Better: A Machine Learning Approach to Corner Detection. IEEE Trans. PAMI 32(1), 105–119 (2010)
30. Harris, C., Stephens, M.: A Combined Corner and Edge Detector. In: Proc. of The Fourth Alvey Vision Conference, pp. 147–151 (1988)
31. Mair, E., Hager, G.D., Burschka, D., Suppa, M., Hirzinger, G.: Adaptive and Generic Corner Detection Based on the Accelerated Segment Test. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part II. LNCS, vol. 6312, pp. 183–196. Springer, Heidelberg (2010)
32. Dahl, A.L., Aanæs, H., Pedersen, K.S.: Finding the Best Feature Detector-Descriptor Combination. In: 3DIMPVT, pp. 318–325 (2011)
33. Fischler, M.A., Bolles, R.C.: Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Communications of the ACM 24(6), 381–395 (1981)
34. Zitnick, L., Krishnan, R.: Edge Foci Interest Points. In: ICCV, pp. 359–366 (2011)
35. Bradski, G.: The OpenCV Library. Dr. Dobb's Journal of Software Tools (2000)
36. Kurz, D., BenHimane, S.: Inertial sensor-aligned visual feature descriptors. In: CVPR, pp. 161–166 (2011)