BRISK: Binary Robust invariant scalable keypoints

Conference Paper *in* Proceedings / IEEE International Conference on Computer Vision. IEEE International Conference on Computer Vision · November 2011

DOI: 10.1109/ICCV.2011.6126542 · Source: DBLP

3 authors:

Stefan Leutenegger
ETH Zurich
88 PUBLICATIONS **9,041** CITATIONS
SEE PROFILE

Margarita Chli
ETH Zurich
62 PUBLICATIONS **6,512** CITATIONS
SEE PROFILE

Roland Siegwart
ETH Zurich
799 PUBLICATIONS **53,450** CITATIONS
SEE PROFILE

BRISK: Binary Robust Invariant Scalable Keypoints

Stefan Leutenegger, Margarita Chli and Roland Y. Siegwart
Autonomous Systems Lab, ETH Zürich
{stefan.leutenegger, margarita.chli, and roland.siegwart}@mavt.ethz.ch

Abstract

Effective and efficient generation of keypoints from an image is a well-studied problem in the literature and forms the basis of numerous Computer Vision applications. Established leaders in the field are the SIFT and SURF algorithms which exhibit great performance under a variety of image transformations, with SURF in particular considered as the most computationally efficient amongst the high-performance methods to date.

In this paper we propose BRISK[1], a novel method for keypoint detection, description and matching. A comprehensive evaluation on benchmark datasets reveals BRISK's adaptive, high quality performance as in state-of-the-art algorithms, albeit at a dramatically lower computational cost (an order of magnitude faster than SURF in cases). The key to speed lies in the application of a novel scale-space FAST-based detector in combination with the assembly of a bit-string descriptor from intensity comparisons retrieved by dedicated sampling of each keypoint neighborhood.

1. Introduction

Decomposing an image into local regions of interest or *'features'* is a widely applied technique in Computer Vision used to alleviate complexity while exploiting local appearance properties. Image representation, object recognition and matching, 3D scene reconstruction and motion tracking all rely on the presence of stable, representative features in the image, driving research and yielding a plethora of approaches to this problem.

The ideal keypoint detector finds salient image regions such that they are *repeatably* detected despite change of viewpoint; more generally it is robust to all possible image transformations. Similarly, the ideal keypoint descriptor captures the most important and distinctive information content enclosed in the detected salient regions, such that the same structure can be *recognized* if encountered. Moreover, on top of fulfilling these properties to achieve the desired quality of keypoints, the speed of detection and description needs also to be optimized to fit within the time-constraints of the task at hand.

In principle, state-of-the-art algorithms target applications with either strict requirements in precision or speed of computation. Lowe's SIFT approach [9] is widely accepted as one of highest quality options currently available, promising distinctiveness and invariance to a variety of common image transformations – however, the at the expense of computational cost. On the other end of the spectrum, a combination of the FAST [14] keypoint detector and the BRIEF [4] approach to description offers a much more suitable alternative for real-time applications. However, despite the clear advantage in speed, the latter approach suffers in terms of reliability and robustness as it has minimal tolerance to image distortions and transformations, in particular to in-plane rotation and scale change. As a result, real-time applications like SLAM [6] need to employ probabilistic methods [5] for data association to discover matching consensus.

The inherent difficulty in extracting suitable features from an image lies in balancing two competing goals: high-quality description and low computational requirements. This is where this work aims to set a new milestone with the BRISK methodology. Perhaps the most relevant work tackling this problem is SURF [2] which has been demonstrated to achieve robustness and speed, only, as evident in our results, BRISK achieves comparable quality of matching at much less computation time. In a nutshell, this paper proposes a novel method for generating keypoints from an image, structured as follows:

- **Scale-space keypoint detection**: Points of interest are identified across both the image and scale dimensions using a saliency criterion. In order to boost efficiency of computation, keypoints are detected in octave layers of the image pyramid as well as in layers in-between. The location and the scale of each keypoint are obtained in the continuous domain via quadratic function fitting.

- **Keypoint description**: A sampling pattern consisting of

[1]The reference implementation of BRISK can be downloaded from http://www.asl.ethz.ch/people/lestefan/personal/BRISK

points lying on appropriately scaled concentric circles is applied at the neighborhood of each keypoint to retrieve gray values: processing local intensity gradients, the feature characteristic direction is determined. Finally, the oriented BRISK sampling pattern is used to obtain pairwise brightness comparison results which are assembled into the binary BRISK descriptor.

Once generated, the BRISK keypoints can be matched very efficiently thanks to the binary nature of the descriptor. With a strong focus on efficiency of computation, BRISK also exploits the speed savings offered in the SSE instruction set widely supported on today's architectures.

2. Related Work

Identifying local interest points to be used for image matching can be traced a long way back in the literature, with Harris and Stephens [7] proposing one of the earliest and probably most well-known corner detectors. The seminal work of Mikolajzyk *et al.*[13] presented a comprehensive evaluation of the most competent detection methods at the time, which revealed no single all-purpose detector but rather the complementary properties of the different approaches depending on the context of the application. The more recent FAST criterion [14] for keypoint detection has become increasingly popular in state-of-the-art methods with hard real-time constraints, with AGAST [10] extending this work for improved performance.

Amongst the best quality features currently in the literature is the SIFT [9]. The high descriptive power and robustness to illumination and viewpoint changes has rated the SIFT descriptor at the top of the rankings list in the survey in [11]. However, the high dimensionality of this descriptor makes SIFT prohibitively slow. PCA-SIFT [8] reduced the descriptor from 128 to 36 dimensions, compromising however its distinctiveness and increasing the time for descriptor formation which almost annihilates the increased speed of matching. The GLOH descriptor [12] is also worth noting here, as it belongs to the family of SIFT-like methods and has been shown to be more distinctive but also more expensive to compute than SIFT.

The growing demand for high-quality, high-speed features has led to more research towards algorithms able to process richer data at higher rates. Notable is the work of Agrawal *et al.*[1] who apply a center-symmetric local binary pattern as an alternative to SIFT's orientation histograms approach. The most recent BRIEF [4] is designed for super-fast description and matching and consists of a binary string containing the results of simple image intensity comparisons at random pre-determined pixel locations. Despite the simplicity and efficiency of this approach, the method is very sensitive to image rotation and scale changes restricting its application to general tasks.

Probably the most appealing features at the moment are the SURF [2], which have been demonstrated to be significantly faster than SIFT. SURF detection uses the determinant of the Hessian matrix (blob detector), while the description is done by summing Haar wavelet responses at the region of interest. While demonstrating impressive timings with respect to the state-of-the-art, SURF are, in terms of speed, still orders of magnitude away from the fastest, yet limited quality features currently available.

In this paper, we present a novel methodology dubbed 'BRISK' for high-quality, fast keypoint detection, description and matching. As suggested by the name, the method is rotation as well as scale invariant to a significant extent, achieving performance comparable to the state-of-the-art while dramatically reducing computational cost. Following a description of the approach, we present experimental results performed on the benchmark datasets and using the standardized evaluation method of [12, 13]. Namely, we present evaluation of BRISK with respect to SURF and SIFT which are widely accepted as a standard of comparison under common image transformations.

3. BRISK: The Method

In this section, we describe the key stages in BRISK, namely feature detection, descriptor composition and keypoint matching to the level of detail that the motivated reader can understand and reproduce. It is important to note that the modularity of the method allows the use of the BRISK detector in combination with any other keypoint descriptor and vice versa, optimizing for the desired performance and the task at hand.

3.1. Scale-Space Keypoint Detection

With the focus on efficiency of computation, our detection methodology is inspired by the work of Mair *et al.*[10] for detecting regions of interest in the image. Their AGAST is essentially an extension for accelerated performance of the now popular FAST, proven to be a very efficient basis for feature extraction. With the aim of achieving invariance to scale which is crucial for high-quality keypoints, we go a step further by searching for maxima not only in the image plane, but also in scale-space using the FAST score s as a measure for saliency. Despite discretizing the scale axis at coarser intervals than in alternative high-performance detectors (*e.g.* the Fast-Hessian [2]), the BRISK detector estimates the true scale of each keypoint in the continuous scale-space.

In the BRISK framework, the scale-space pyramid layers consist of n octaves c_i and n intra-octaves d_i, for $i = \{0, 1, \ldots, n-1\}$ and typically $n = 4$. The octaves are formed by progressively half-sampling the original image (corresponding to c_0). Each intra-octave d_i is located in-between layers c_i and c_{i+1} (as illustrated in Figure

1). The first intra-octave d_0 is obtained by downsampling the original image c_0 by a factor of 1.5, while the rest of the intra-octave layers are derived by successive halfsampling. Therefore, if t denotes scale then $t(c_i) = 2^i$ and $t(d_i) = 2^i \cdot 1.5$.

It is important to note here that both FAST and AGAST provide different alternatives of mask shapes for keypoint detection. In BRISK, we mostly use the 9-16 mask, which essentially requires at least 9 consecutive pixels in the 16-pixel circle to either be sufficiently brighter or darker than the central pixel for the FAST criterion to be fulfilled.

Initially, the FAST 9-16 detector is applied on each octave and intra-octave separately using the same threshold T to identify potential regions of interest. Next, the points belonging to these regions are subjected to a non-maxima suppression in scale-space: firstly, the point in question needs to fulfill the maximum condition with respect to its 8 neighboring FAST scores s in the same layer. The score s is defined as the maximum threshold still considering an image point a corner. Secondly, the scores in the layer above and below will need to be lower as well. We check inside equally sized square patches: the side-length is chosen to be 2 pixels in the layer with the suspected maximum. Since the neighboring layers (and therefore its FAST scores) are represented with a different discretization, some interpolation is applied at the boundaries of the patch. Figure 1 depicts an example of this sampling and the maxima search.

The detection of maxima across the scale axis at octave c_0 is a special case: in order to obtain the FAST scores for a virtual intra-octave d_{-1} below c_0, we apply the FAST 5-8 mask on c_0. However, the scores in patch of d_{-1} are in this case not required to be lower than the score of the examined point in octave c_0.

Considering image saliency as a continuous quantity not only across the image but also along the scale dimension, we perform a sub-pixel and continuous scale refinement for each detected maximum. In order to limit complexity of the refinement process, we first fit a 2D quadratic function in the least-squares sense to each of the three scores-patches (as obtained in the layer of the keypoint, the one above, and the one below) resulting in three sub-pixel refined saliency maxima. In order to avoid resampling, we consider a 3 by 3 score patch on each layer. Next, these refined scores are used to fit a 1D parabola along the scale axis yielding the final score estimate and scale estimate at its maximum. As a final step, we re-interpolate the image coordinates between the patches in the layers next to the determined scale. An example of the BRISK detection in two images of the Boat sequence (defined in Section 4) is shown up-close in Figure 2.

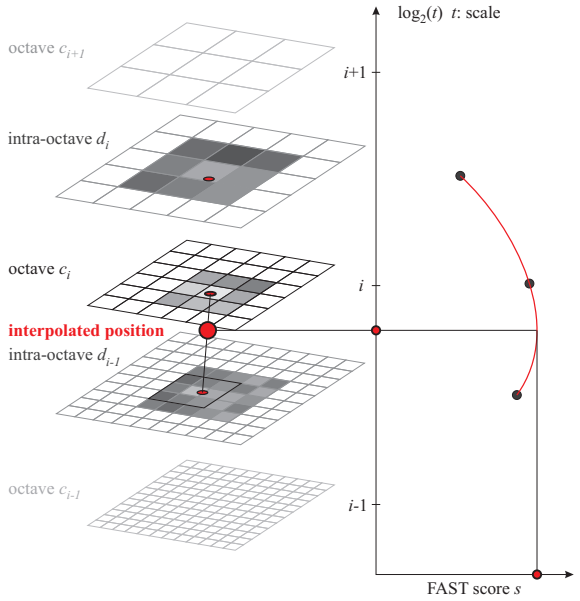
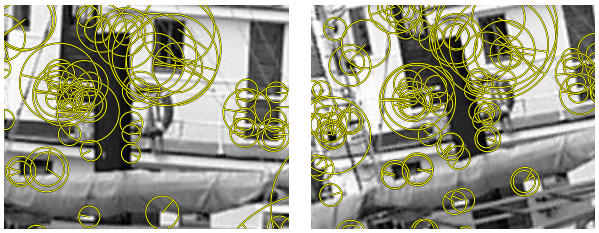


Figure 1. Scale-space interest point detection: a keypoint (*i.e.* saliency maximum) is identified at octave c_i by analyzing the 8 neighboring saliency scores in c_i as well as in the corresponding scores-patches in the immediately-neighboring layers above and below. In all three layers of interest, the local saliency maximum is sub-pixel refined before a 1D parabola is fitted along the scale-axis to determine the true scale of the keypoint. The location of the keypoint is then also re-interpolated between the patch maxima closest to the determined scale.



(a) Boat image 1 (b) Boat image 2

Figure 2. Close-up of a BRISK detection example on images 1 and 2 of the Boat sequence exhibiting small zoom and in-plane rotation. The size of the circles denote the scale of the detected keypoints while the radials denote their orientation. For clarity, the detection threshold is set here to a stricter value than in the typical setup, yielding slightly lower repeatability.

3.2. Keypoint Description

Given a set of keypoints (consisting of sub-pixel refined image locations and associated floating-point scale values), the BRISK descriptor is composed as a binary string by concatenating the results of simple brightness comparison tests. This idea has been demonstrated in [4] to be very efficient, however here we employ it in a far more qualitative manner. In BRISK, we identify the characteristic direction of each keypoint to allow for orientation-normalized descriptors and hence achieve rotation invariance which is key to general robustness. Also, we carefully select the brightness comparisons with the focus on maximizing descriptiveness.

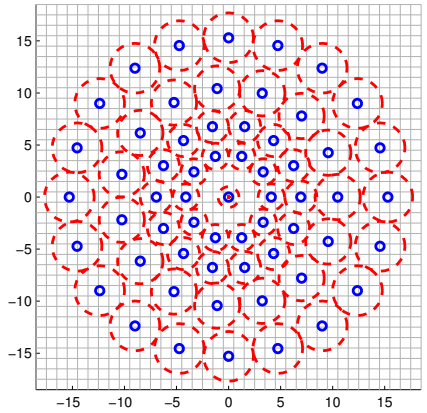


Figure 3. The BRISK sampling pattern with $N = 60$ points: the small blue circles denote the sampling locations; the bigger, red dashed circles are drawn at a radius σ corresponding to the standard deviation of the Gaussian kernel used to smooth the intensity values at the sampling points. The pattern shown applies to a scale of $t = 1$.

3.2.1 Sampling Pattern and Rotation Estimation

The key concept of the BRISK descriptor makes use of a pattern used for sampling the neighborhood of the keypoint. The pattern, illustrated in Figure 3, defines N locations equally spaced on circles concentric with the keypoint. While this pattern resembles the DAISY descriptor [15], it is important to note that its use in BRISK is entirely different, as DAISY was built specifically for dense matching, deliberately capturing more information and thus resulting to demanding speed and storage requirements.

In order to avoid aliasing effects when sampling the image intensity of a point $\mathbf{p}_i$ in the pattern, we apply Gaussian smoothing with standard deviation σ_i proportional to the distance between the points on the respective circle. Positioning and scaling the pattern accordingly for a particular keypoint k in the image, let us consider one of the $N \cdot (N-1)/2$ sampling-point pairs $(\mathbf{p}_i, \mathbf{p}_j)$. The smoothed intensity values at these points which are $I(\mathbf{p}_i, \sigma_i)$ and $I(\mathbf{p}_j, \sigma_j)$ respectively, are used to estimate the local gradient $\mathbf{g}(\mathbf{p}_i, \mathbf{p}_j)$ by

$$\mathbf{g}(\mathbf{p}_i, \mathbf{p}_j) = (\mathbf{p}_j - \mathbf{p}_i) \cdot \frac{I(\mathbf{p}_j, \sigma_j) - I(\mathbf{p}_i, \sigma_i)}{\|\mathbf{p}_j - \mathbf{p}_i\|^2}. \quad (1)$$

Considering the set $\mathcal{A}$ of all sampling-point pairs:

$$\mathcal{A} = \left\{ (\mathbf{p}_i, \mathbf{p}_j) \in \mathbb{R}^2 \times \mathbb{R}^2 \mid i < N \wedge j < i \wedge i, j \in \mathbb{N} \right\} \quad (2)$$

we define a subset of short-distance pairings $\mathcal{S}$ and another subset of L long-distance pairings $\mathcal{L}$:

$$\begin{aligned} \mathcal{S} &= \{ (\mathbf{p}_i, \mathbf{p}_j) \in \mathcal{A} \mid \|\mathbf{p}_j - \mathbf{p}_i\| < \delta_{max} \} \subseteq \mathcal{A} \\ \mathcal{L} &= \{ (\mathbf{p}_i, \mathbf{p}_j) \in \mathcal{A} \mid \|\mathbf{p}_j - \mathbf{p}_i\| > \delta_{min} \} \subseteq \mathcal{A}. \end{aligned} \quad (3)$$

The threshold distances are set to $\delta_{max} = 9.75t$ and $\delta_{min} = 13.67t$ (t is the scale of k). Iterating through the point pairs in $\mathcal{L}$, we estimate the overall characteristic pattern direction of the keypoint k to be:

$$\mathbf{g} = \begin{pmatrix} g_x \\ g_y \end{pmatrix} = \frac{1}{L} \cdot \sum_{(\mathbf{p}_i, \mathbf{p}_j) \in \mathcal{L}} \mathbf{g}(\mathbf{p}_i, \mathbf{p}_j). \quad (4)$$

The long-distance pairs are used for this computation, based on the assumption that local gradients annihilate each other and are thus not necessary in the global gradient determination – this was also confirmed by experimenting with variation of the distance threshold δ_{min}.

3.2.2 Building the Descriptor

For the formation of the rotation- and scale-normalized descriptor, BRISK applies the sampling pattern rotated by $\alpha = \arctan2\,(g_y, g_x)$ around the keypoint k. The bit-vector descriptor d_k is assembled by performing all the short-distance intensity comparisons of point pairs $(\mathbf{p}_i^\alpha, \mathbf{p}_j^\alpha) \in \mathcal{S}$ (*i.e.* in the rotated pattern), such that each bit b corresponds to:

$$\begin{aligned} b &= \begin{cases} 1, & I(\mathbf{p}_j^\alpha, \sigma_j) > I(\mathbf{p}_i^\alpha, \sigma_i) \\ 0, & \text{otherwise} \end{cases} \\ &\forall (\mathbf{p}_i^\alpha, \mathbf{p}_j^\alpha) \in \mathcal{S} \end{aligned} \quad (5)$$

While the BRIEF descriptor is also assembled via brightness comparisons, BRISK has some fundamental differences apart from the obvious pre-scaling and pre-rotation of the sampling pattern. Firstly, BRISK uses a deterministic sampling pattern resulting in a uniform sampling-point density at a given radius around the keypoint. Consequently, the tailored Gaussian smoothing will not accidentally distort the information content of a brightness comparison by blurring two close sampling-points in a comparison. Furthermore, BRISK uses dramatically fewer sampling-points than pairwise comparisons (*i.e.* a single point participates in more comparisons), limiting the complexity of looking-up intensity values. Finally, the comparisons here are restricted spatially such that the brightness variations are only required to be locally consistent. With the sampling pattern and the distance thresholds as shown above, we obtain a bit-string of length 512. The bit-string of BRIEF64 also contains 512 bits, thus the matching for a descriptor pair will be performed equally fast by definition.

3.3. Descriptor Matching

Matching two BRISK descriptors is a simple computation of their Hamming distance as done in BRIEF [4]: the number of bits different in the two descriptors is a measure of their dissimilarity. Notice that the respective operations reduce to a bitwise XOR followed by a bit count, which can both be computed very efficiently on today's architectures.

3.4. Notes on Implementation

Here, we give a very brief overview of some implementation issues which contribute significantly to the overall computational performance and the reproducibility of the method. All the BRISK functionality builds on the common 2D feature interface of OpenCV 2.2 allowing easy integration and interchangeability with existing features (SIFT, SURF, BRIEF, etc.).

The detection process uses the AGAST implementation [10] for computing saliency scores. The non-maxima suppression benefits from early termination capability limiting the saliency scores calculation to a minimum. Building the image pyramid makes use of some SSE2 and SSSE3 commands, both concerning the halfsampling as well as the downsampling by a factor of 1.5.

In order to efficiently retrieve gray values with the sampling pattern, we generate a look-up table of discrete rotated and scaled BRISK pattern versions (consisting of the sampling-point locations and the properties of the Gaussian smoothing kernel as well as the indexing of long and short distance pairings) consuming around 40MB of RAM – which is still acceptable for applications constrained to low computational power.

We furthermore use the integral image along with a simplified Gaussian kernel version inspired by [2]: the kernel is scalable when changing σ without any increase in computational complexity. In our final implementation we use as an approximation a simple square box mean filter with floating point boundaries and side length $\rho = 2.6 \cdot \sigma$.

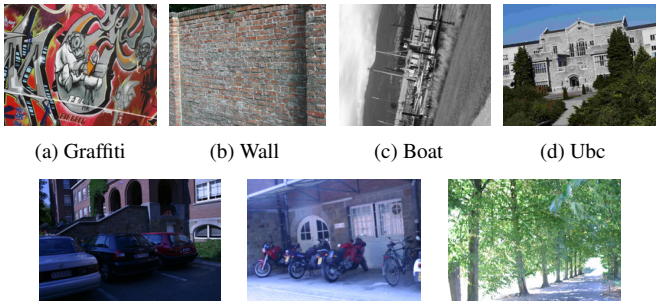
Thus we do not need time-consuming Gaussian smoothing of the whole image with many different kernels, but we instead retrieve single values using an arbitrary parameter σ.

We also integrated an improved SSE Hamming distance calculator achieving matching at 6 times the speed of the current OpenCV implementation as used for example with BRIEF in OpenCV.

4. Experiments

Our proposed method has been extensively tested following the now established evaluation method and datasets in the field first proposed by Mikolajczyk and Schmid [12, 13]. For the sake of consistency with results presented in other works, we also used their MATLAB evaluation scripts which are available online. Each of the datasets contains a sequence of six images exhibiting an increasing amount of transformation. All comparisons here are performed against the first image in each dataset. Figure 4 shows one image for each dataset analyzed.

The transformations cover view-point change (Graffiti and Wall), zoom and rotation (Boat), blur (Bikes and Trees), brightness changes (Leuven) as well as JPEG compression (Ubc). Since the viewpoint change scenes are planar, the image pairs in all sequences are provided with a ground truth homography used to determine the corresponding keypoints. In the rest of the section we present quantitative results concerning the detector and descriptor performance of BRISK compared to SIFT (OpenCV2.2 implementation) as well as SURF (original implementation). Our evaluation uses similarity matching which considers any pair of keypoints with descriptor distance below a certain threshold a match – in contrast to *e.g.* nearest neighbor matching, where a database is searched for the match with the lowest descriptor distance. Finally, we also demonstrate BRISK's big advantage in computational speed by listing comparative timings.



(a) Graffiti (b) Wall (c) Boat (d) Ubc

(e) Leuven (f) Bikes (g) Trees

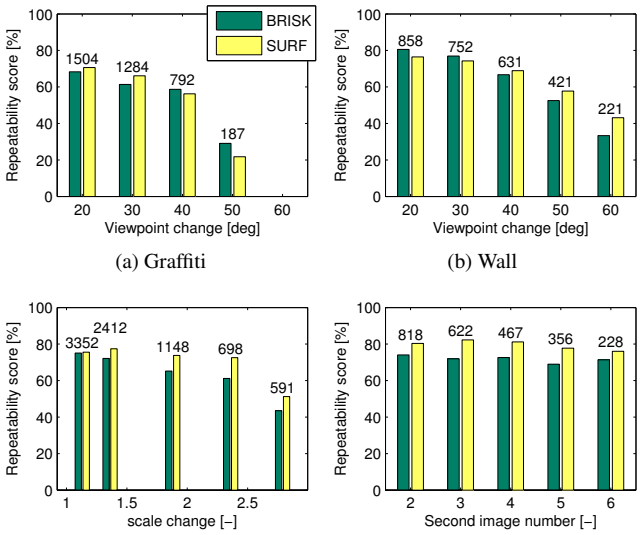
Figure 4. Datasets used for evaluation: viewpoint change (Graffiti and Wall), zoom and rotation (Boat), JPEG compression (Ubc), brightness change (Leuven), and blur (Bikes and Trees).

4.1. BRISK Detector Repeatability

The detector repeatability score as defined in [13] is calculated as the ratio between the corresponding keypoints and the minimum total number of keypoints visible in both images. The correspondences are identified by looking at the overlap area of the keypoint region in one image (*i.e.* the extracted circle) and the projection of the keypoint region from the other image (*i.e.* ellipse-like): if the region of intersection is larger than 50% of the union of the two regions, it is considered a correspondence. Note that this method is largely dependent on the assignment of the keypoint circle radius, *i.e.* the constant factor between scale and radius. We choose this such that the average radii obtained with the BRISK detector approximately match the average radii obtained with the SURF and SIFT detectors.

The assessment of repeatability scores (a selection of results is shown in Figure 5) is performed using constant BRISK detection thresholds across one sequence. For the sake of a fair comparison with the SURF detector, we adapt the respective Hessian threshold such that it outputs approximately the same number of correspondences in the similarity based matching setup.

As illustrated in Figure 5, the BRISK detector exhibits



(a) Graffiti (b) Wall

(c) Boat (d) Leuven

Figure 5. Repeatability scores for 50% overlap error of the BRISK and the SURF detector. The resulting similarity correspondences (approximately matched between the detectors) are given as numbers above the bars.

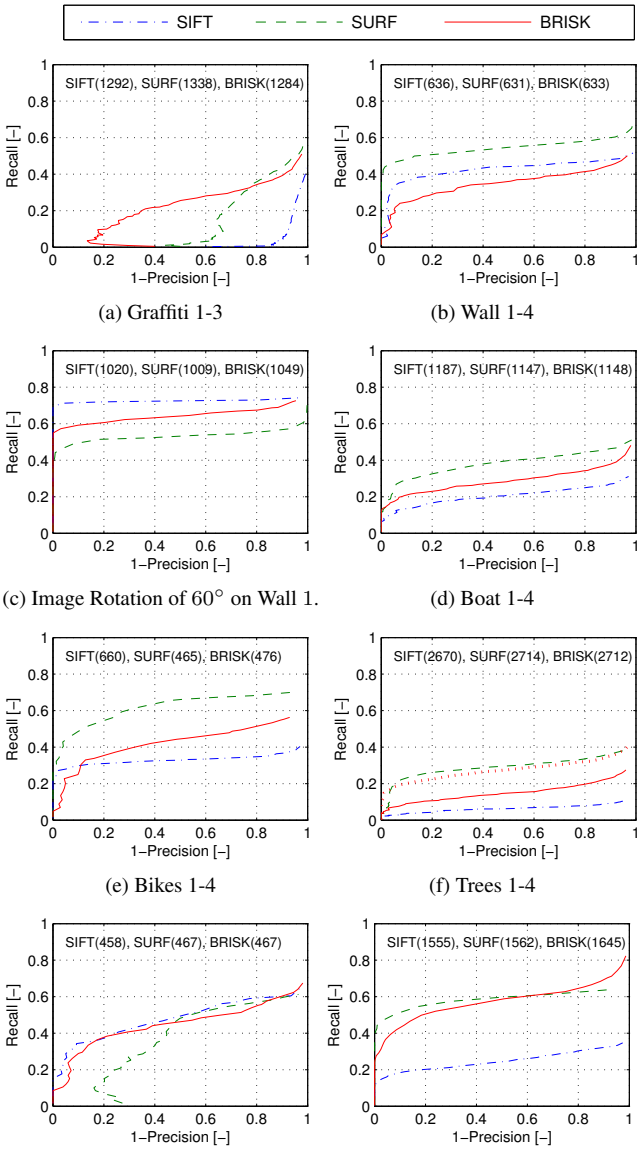
equivalent repeatability as the SURF detector as long as the image transformations applied are not too large. Given the clear advantage in computational cost of the BRISK over the SURF detector however, the proposed method constitutes a strong competitor, even if the performance at larger transformations appears to be slightly inferior.

4.2. Evaluation and Comparison of the Overall BRISK Algorithm

Since our work aims at providing an overall fast as well as robust detection, description and matching, we evaluate the joint performance of all these stages in BRISK and compare it to SIFT and SURF. Figure 6 shows the precision-recall curves using threshold-based similarity matching for a selection of image pairs of different datasets. Again, for this assessment we adapt the detection thresholds such that they output an approximately equal number of correspondences in the spirit of fairness. Note that the evaluation results here are different from the ones in [3], where all descriptors are extracted on the same regions (obtained with the Fast-Hessian detector).

As illustrated in Figure 6, BRISK performs competitively with SIFT and SURF in all datasets and even outperforms the other two in some cases. The reduced performance of BRISK in the Trees dataset is attributed to the detector performance: while SURF detects 2606 and 2624 regions in the images, respectively, BRISK only detects 2004 regions in image 4 compared to 5949 found in image 1 to achieve the approximately same number of correspondences. The same holds for the other blur dataset, Bikes: saliency as assessed with FAST is inherently more sensitive to blur than blob-like detectors. We therefore also show the evaluation of the BRISK descriptors extracted from the SURF regions for the Trees dataset, demonstrating again that the descriptor performance is comparable to SURF.

Evidently, SIFT performs significantly worse in the Trees, Boat, and Ubc datasets, which can be explained with the limited detector repeatability in these cases. On the



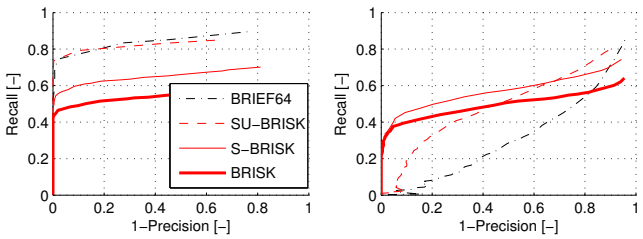
(a) Graffiti 1-3 (b) Wall 1-4

(c) Image Rotation of 60° on Wall 1. (d) Boat 1-4

(e) Bikes 1-4 (f) Trees 1-4

(g) Leuven 1-4 (h) Ubc 1-4

Figure 6. Evaluation results showing precision-recall curves (of all detection, extraction and matching stages jointly) for BRISK, SURF and SIFT. Results are shown for viewpoint changes (a and b), pure in-plane rotation (c), zoom and rotation (d), blur (e and f), brightness changes (g) and JPEG compression (h). The number of similarity correspondences are indicated in the figures per algorithm. The red dotted line in (f) shows the performance of BRISK descriptors extracted from SURF regions, yielding 2274 correspondences. Overall, BRISK exhibits competitive performance in all cases and even outperforms SIFT and SURF in some cases.



(a) Wall 1-2 (b) Boat 1-2

Figure 7. Comparison of different BRISK versions to 64 byte BRIEF. BRIEF, as well as both SU-BRISK (single-scale, unrotated) and S-BRISK (single-scale) are extracted from AGAST keypoints detected in the original image. Notice that the BRISK pattern was scaled such that it matches the BRIEF patch size. The standard version of BRISK had to be extracted from our scale-invariant corner detection with adapted threshold to match the number of correspondences: they are 850 in the Wall pair and 1530 in the Boat pair.

other hand, SIFT and BRISK handle the important case of pure in-plane rotation very well and better than SURF.

In order to complete the experimental section, we want to make the link to BRIEF. Figure 7 shows a comparison of the unrotated, single-scale BRISK version (SU-BRISK) to 64 byte BRIEF features on the same (single scale) AGAST keypoints. Also included are the rotation invariant, single-scale S-BRISK, as well as the standard BRISK. The experiment is conducted with two image pairs: on the one hand, we used the first two images in the Wall dataset proving that SU-BRISK and BRIEF64 are exhibiting a very similar performance in the absence of scale change and in-plane rotation. Notice that this is really the situation BRIEF was designed for. On the other hand, we applied the different versions to the first two images of the Boat sequence: this experiment demonstrates some advantage of the SU-BRISK over BRIEF in terms of robustness against small rotation (10°) and scale changes (10%). Furthermore, the well known and intuitive price for both rotation and scale invariance is easily observable.

4.3. Timings

Timings have been recorded on a laptop with a quad-core i7 2.67 GHz processor (only using one core, however) running Ubuntu 10.04 (32-bit), using the implementation and setup as detailed above. Table 1 presents the results concerning detection on the first image of the Graffiti sequence, while Table 2 shows the matching times. The values are averaged over 100 runs. Note that all matchers do a brute-force descriptor distance computation without any early termination optimizations.

The timings show a clear advantage of BRISK. Its detection and descriptor computation is typically an order of magnitude faster than the one of SURF, which are considered to be the fastest rotation and scale invariant features currently available. It is also important to highlight that

	SIFT	SURF	BRISK
Detection threshold	4.4	45700	67
Number of points	1851	1557	1051
Detection time [ms]	1611	107.9	17.20
Description time [ms]	9784	559.1	22.08
Total time [ms]	11395	667.0	39.28
Time per point (ms)	**6.156**	**0.4284**	**0.03737**

Table 1. Detection and extraction timings for the first image in the Graffiti sequence (size: 800 × 640 pixels).

	SIFT	SURF	BRISK
Points in first image	1851	1557	1051
Points in second image	2347	1888	1385
Total time [ms]	291.6	194.6	29.92
Time per comparison [ns]	**67.12**	**66.20**	**20.55**

Table 2. Matching timings for the Graffiti image 1 and 3 setup.

BRISK is easily scalable for faster execution by reducing the number of sampling-points in the pattern at some expense of matching quality – which might be affordable in a particular application. Moreover, scale and/or rotation invariance can be omitted trivially, increasing the speed as well as the matching quality in applications where they are not needed.

4.4. An Example

Complementary to the extensive evaluation presented above, we also provide a real-world example demonstrating matching using BRISK. Figure 8 shows an image pair exhibiting various transformations. A similarity match with a threshold of 90 was performed (out of 512 comparisons) resulting in robust matches without significant outliers.

5. Conclusions

We have presented a novel method named BRISK, which tackles the classic Computer Vision problem of detecting, describing and matching image keypoints for cases without sufficient *a priori* knowledge on the scene and camera poses. In contrast to well-established algorithms with proven high performance, such as SIFT and SURF, the method at hand offers a dramatically faster alternative at comparable matching performance – a statement which we base on an extensive evaluation using an established framework. BRISK relies on an easily configurable circular sampling pattern from which it computes brightness comparisons to form a binary descriptor string. The unique properties of BRISK can be useful for a wide spectrum of applications, in particular for tasks with hard real-time constraints or limited computation power: BRISK finally offers the quality of high-end features in such time-demanding applications.

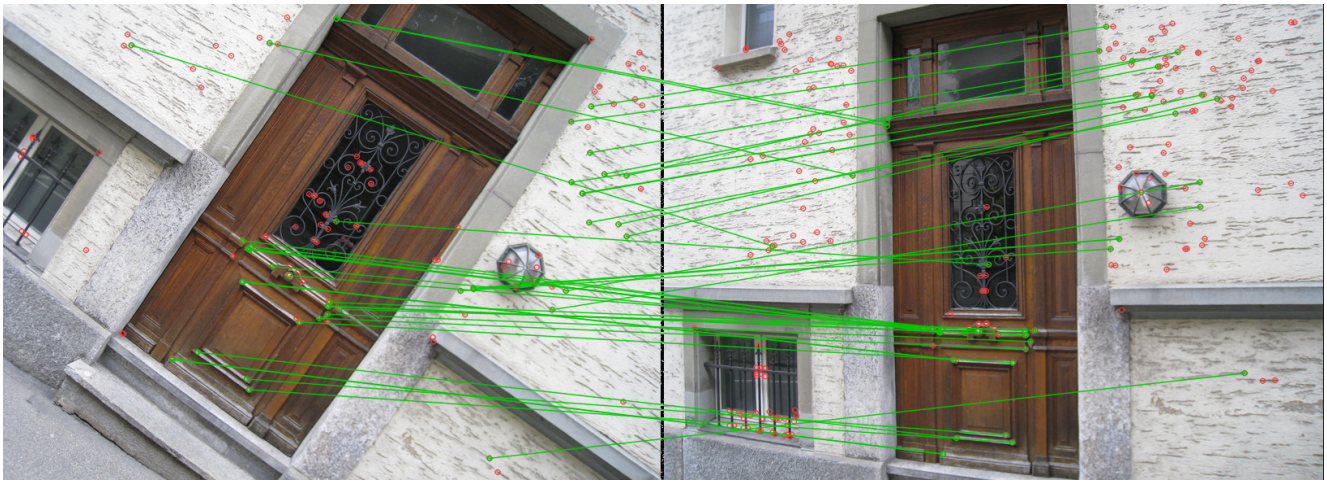


Figure 8. BRISK matching example: a detection threshold of 70 is used and a matching Hamming distance threshold of 90. The resulting matches are connected by the green lines showing no clear false positives. The authors provide a reference implementation of BRISK downloadable from http://www.asl.ethz.ch/people/lestefan/personal/BRISK .

Amongst avenues for further research into BRISK, we aim to explore alternatives to the scale-space maxima search of saliency scores to yield higher repeatability whilst maintaining speed. Furthermore, we aim at analyzing both theoretically and experimentally the BRISK pattern and the configuration of comparisons, such that the information content and/or robustness of the descriptor is maximized.

6. Acknowledgements

This research was supported by the Autonomous Systems Lab, ETH Zurich and the EC's 7th Framework Programme (FP7/2001-2013) under grant agreement no. 231855 (sFly). We are grateful to Simon Lynen and Davide Scaramuzza for their valuable inputs, as well as to many other colleagues at ETH Zurich for very helpful discussions.

References

[1] M. Agrawal, K. Konolige, and M. R. Blas. CenSurE: Center surround extremas for realtime feature detection and matching. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2008. 2

[2] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. SURF: Speeded up robust features. *Computer Vision and Image Understanding (CVIU)*, 110(3):346–359, 2008. 1, 2, 5

[3] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2006. 6

[4] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. BRIEF: Binary Robust Independent Elementary Features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010. 1, 2, 3, 4

[5] M. Chli and A. J. Davison. Active Matching. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2008. 1

[6] A. J. Davison, N. D. Molton, I. Reid, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 29(6):1052–1067, 2007. 1

[7] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of 4th Alvey Vision Conference*, pages 147–151, 1988. 2

[8] Y. Ke and R. Sukthankar. PCA-SIFT: A More Distinctive Representation for Local Image Descriptors. 2004. 2

[9] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 60(2):91–110, 2004. 1, 2

[10] E. Mair, G. D. Hager, D. Burschka, M. Suppa, and G. Hirzinger. Adaptive and generic corner detection based on the accelerated segment test. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010. 2, 5

[11] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003. 2

[12] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2:1115–1125, 2005. 2, 5

[13] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Gool. A comparison of affine region detectors. *International Journal of Computer Vision (IJCV)*, 65(1):43–72, 2005. 2, 5

[14] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2006. 1, 2

[15] E. Tola, V. Lepetit, and P. Fua. Daisy: an Efficient Dense Descriptor Applied to Wide Baseline Stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 32(5):815–830, 2010. 4