Evaluation of low-complexity visual feature detectors and descriptors

A. Canclini, M. Cesana, A.Redondi, M.Tagliasacchi
Politecnico di Milano
P.zza Leonardo da Vinci 32, Milano, Italy
{canclini,cesana,redondi,tagliasa}@polimi.it

J. Ascenso, R. Cilla
Instituto Superior de Engenharia de Lisboa
Instituto de Telecomunicações
Av. Rovisco Pais, 1, Lisbon, Portugal
joao.ascenso@lx.it.pt, rodri.cilla@lx.it.pt

***Abstract*—Several visual feature extraction algorithms have recently appeared in the literature, with the goal of reducing the computational complexity of state-of-the-art solutions (e.g., SIFT and SURF). Therefore, it is necessary to evaluate the performance of these emerging visual descriptors in terms of processing time, repeatability and matching accuracy, and whether they can obtain competitive performance in applications such as image retrieval. This paper aims to provide an up-to-date detailed, clear, and complete evaluation of local feature detector and descriptors, focusing on the methods that were designed with complexity constraints, providing a much needed reference for researchers in this field. Our results demonstrate that recent feature extraction algorithms, e.g., BRISK and ORB, have competitive performance requiring much lower complexity and can be efficiently used in low-power devices.**

***Index Terms*—local feature detectors, local feature descriptors, binary descriptors, image retrieval**

I. Introduction

Visual features are routinely adopted in several applications, ranging from image/video retrieval, object recognition and tracking, visual odometry, etc. Visual features are often related to early vision, e.g., shape and appearance, and can be divided in two broad classes: global and local features. Global features represent the content of the image as a whole, without direct reference to the spatial layout of the visual content. Typically, they lead to fixed-dimensional feature vectors that can be efficiently indexed and matched, thus being particularly suited for large scale scenarios. However, global features are not suitable in those scenarios that require a more precise description of the local content of an image. To overcome this limitation, local features attempt to find a representation that is robust to occlusions, geometric transformations and illumination changes. Local features are extracted according to a two-step process. First, a detector analyzes the image to extract a set of salient keypoints, e.g., interest points that represent the most informative parts of an image. In general, the number of extracted keypoints is content dependent. Then, image patches around each keypoint are processed further and compactly represented by means of fixed-dimensional descriptors that capture, e.g., their photometric properties.

The project GreenEyes acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open grant number:296676.

The detection and extracting of local features is a time-consuming task. When feature extraction is performed on low-power devices, e.g., nodes in a wireless visual sensor network [**?**], it is particularly interesting to consider computationally efficient algorithms for local feature detectors and descriptors. The main contribution of this work is a comprehensive evaluation of a set of representative detectors and descriptors available in the literature, whose design was guided by computational efficiency concerns. While specific detector-descriptor pairs are often jointly considered, we make the effort of distinguishing the two functional components in our evaluation. Hence, in Section II local feature detectors are considered. Both blob-like (SIFT - DoG [**?**], SURF - fast Hessian [**?**], CenSure [**?**]) and corner-like (FAST [**?**], AGAST [**?**], BRISK [**?**], ORB [**?**]) detectors were included in our study, which reports computational complexity, expressed in terms of processing time, as well as keypoint repeatibility. Then, Section II focus on visual descriptors, analyzing both non-binary (SIFT [**?**], SURF [**?**]) and binary (BRIEF [**?**], ORB [**?**], BRISK [**?**], FREAK [**?**]) descriptors. Also in this case we report the computational complexity in terms of processing time. Morever, matching accuracy is evaluated in terms of Receiver Operating Characteristic (ROC) curves, which were obtained using the dataset in [**?**], providing a ground truth of matching and non-matching image patches. Finally, in Section **??**, the detectors and descriptors are evaluated for the image retrieval scenario (one of the many scenarios in which visual features can be used).

In the past, other works have addressed the performance evaluation of local features to compare the performance of different detectors or descriptors. For example, in [**?**], local features detector were considered, with the goal of highlighting the properties of affine-invariat detectors. In [**?**], a comprehensive evaluation of feature descriptors is reported. However, these references are partially outdated since they do not consider the most recent advances that led to computationally efficient algorithms for local feature extraction. Therefore, this paper provides an up-to-date evaluation that includes current state-of-the-art detectors and descriptor, especially considering detectors and descriptors for resource-constrained devices.

II. Detector Evaluation

This Section summarizes the keypoints detectors used in this evaluation. The detectors can be organized in two main

classes, namely blob-like feature detectors and corner-like feature detectors. For each class, detectors are presented in chronological order of appearance in the literature.

A. Blob Detectors

Blob detectors detect local extrema of the responses of particular filters as keypoints. These filters are generally designed to be approximations of the Laplacian of Gaussian (LoG), which was shown to be scale invariant in [**?**].

1) SIFT: The Scale Invariant Feature Transform (SIFT) [**?**] is widely recognized as the "gold-standard" approach for scale and rotation invariant interest point detection. The SIFT detector is implemented using Differences of Gaussians (DoG), an approximation of the LoG and requires the computation of a scale-space pyramid. Such pyramid is composed by a number of layers that can be referred to as octaves and scales, the latter located in-between the former. In SIFT, each octave is obtained by progressively half-sampling the original image, while each scale is obtained by convolving the corresponding octave with Gaussians functions that have incremental scales. Then, adjacent layers are subtracted and the DoG pyramid is produced. The keypoints are identified as maxima or minima of the DoG at each scale. An additional non-maxima suppression step is done by comparing the candidate keypoints with the $3 \times 3 \times 3$ neighboring points in the pixel-scale volume. SIFT also provides rejection of keypoints with low constrast or that are poorly localized along an edge.

2) SURF: The Speeded Up Robust Feature (SURF) [**?**] detector is built around the multi-scale Fast Hessian detector, which approximates the DoG approach with much lower computational effort. The SURF detector is based on the determinant of the Hessian matrix. Since the computation of the Hessian matrix implies convolutions with Gaussian second order derivatives, which can be very costly, SURF approximates them with box filters that can be computed efficiently using integral images. The approximated Hessian determinant is

$$c(x, y, \sigma) = D_{xx}(\sigma) \cdot D_{yy}(\sigma) - [0.9 D_{xy}(\sigma)]^2 \,, \qquad (1)$$

where D_{xx}, D_{xy} and D_{yy} are the approximated convolutions obtained by using the box filters. A pixel (x, y) is marked as a keypoint at scale σ if $c(x, y, \sigma) > \theta$, where θ is a fixed threshold value. As in SIFT, non-maxima suppression is applied in a $3 \times 3 \times 3$ neighborhood.

3) CenSurE: The Center-Surround Extrema (CenSurE) approach [**?**] is another approximation of the LoG filter, which uses box filters and integral images in an efficient way as in SURF. The main difference with its SIFT and SURF predecessors is that features are detected at all scales at every pixel in the original image, i.e., without half sampling each octave. This results in features which are more stable at higher level of the scale-space pyramids, and outperforms other approaches, in particular for the task of visual odometry. Also in this case, non-maxima and edge suppression is applied in a similar fashion as in SIFT and SURF.

B. Corner Detectors

1) FAST: The Fast Accelerated Segment Test (FAST) [**?**] detector represents a clear breakthrough in high-speed corner detectors and it is based on the Accelerated Segment Test (AST). AST classifies a candidate point p (with intensity I_p) as a corner if n contiguous pixels in the Bresenham circle of radius 3 around p are all brighter than $I_p + t$, or all darker than $I_p - t$, with t a predefined threshold. Each corner is then given a score s, defined as the largest threshold for which p is classified as a corner. Additionally, the authors presented a machine learning approach to create decision trees that allows FAST to classify a candidate point with only a few pixel tests, thus speeding-up the detection process. This solution requires, on average, less than 2.3 tests per pixel to determine whether or not it is a feature.

2) AGAST: In AGAST (Adaptive and Generic AST) [**?**], the performance of FAST is increased by changing the way in which the decision tree is created. Instead of using a fixed decision tree to classify a point, AGAST provides dynamic adaptation of the decision tree based on the image section currently processed. The authors show that, in a controlled scenario, a speed-up of almost 50% can be achieved with respect to FAST.

3) BRISK: A drawback of FAST and AGAST is that they do not produce scale-invariant keypoints, which are crucial for several applications. In BRISK (Binary Robust Invariant Scalable Keypoints) [**?**], this gap is closed by searching corners not only in the original image plane, but also in the scale-space. Mimicking the SIFT scale-space analysis, in BRISK, the AGAST detector is executed for each pyramid layer separately and non-maxima suppression is applied in the scale-space: a point p is classified as a corner only if its score s is greater than all its neighboring pixels in the same layer and in the layer above and below.

4) ORB: Similar to BRISK, Oriented BRIEF (ORB) [**?**] is a refined, scale-invariant version of FAST. In addition, the detector is modified to estimate the orientation of a keypoint, which is later used in the description phase to provide robustness to rotation.

C. Experiments

In this paragraph we report the results of a comparative analysis of the keypoint detection algorithms described in Sections II-A and II-B. For the FAST, CenSurE, ORB, SURF and SIFT detectors, the OpenCV 2.4.4 implementation (the most recent release at the time this paper was written) was used, while for AGAST and BRISK detectors, the original implementations were used. All the experiments have been carried on a personal computer powered by an Intel® i7-3770 CPU at 3.40 GHz, with 8 GB of RAM. All tests were executed on a single core.

1) Processing Time: As a first experiment, the keypoint detection processing time was evaluated. In general, the processing time depends on two factors: i) the number N of detected keypoints, which can be varied by suitably tuning the detection threshold of each detector; and ii) the size

of the input image. For the evaluation, $L = 10$ images of size 1024×768 pixels, randomly selected from the *5K Oxford Buildings* database[1] were considered. Each image was progressively resized from its original size to a minimum size (20% of the original size). An exhaustive evaluation has been performed for each detector, covering the space $(100 \leq N \leq 1800)$ for all the image sizes from 20% to 100%. For reasons of space, we do not report the complete set of results, which are available at [**?**].

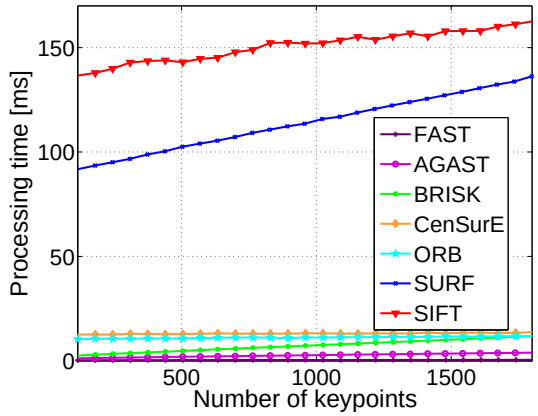


Fig. 1. Processing time as a function of the number of keypoints.

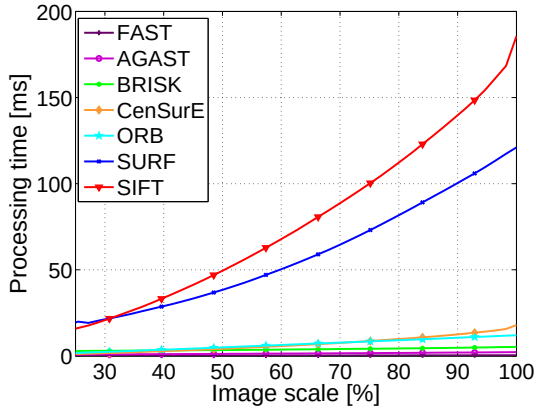


Fig. 2. Processing time as a function of the image size.

Figure **??** shows the processing time, averaged for the L full-size images, as a function of the number of detected keypoints. As shown, for the SIFT, SURF, BRISK, AGAST and FAST detectors the processing time T increases linearly with the number of keypoints N, while CenSurE and ORB processing times are constant. FAST and AGAST achieve the best performances ($T < 2\,\text{ms}$ for $N = 100$ and $T < 4\,\text{ms}$ for $N = 1800$), immediately followed by BRISK ($T \approx 2.5\,\text{ms}$ for $N = 100$ and $T \approx 12\,\text{ms}$ for $N = 1800$), ORB ($T \approx 11\,\text{ms}\ \forall N$) and CenSurE ($T \approx 13\,\text{ms}\ \forall N$). SIFT and SURF are sensibly more time-consuming than all the other detectors. In particular, SURF ranges from $T \approx 90\,\text{ms}$ for $N = 100$ to $T \approx 135\,\text{ms}$ for $N = 1800$; SIFT exhibits the highest processing time (from $T \approx 135\,\text{ms}$ to $T \approx 165\,\text{ms}$).

In Figure **??** the average processing time is shown as a function of the image size. Here, the number of detected keypoints is kept fixed to $N = 500$. For all the detectors, the processing time is a quadratic function of the image size, which means that complexity of detection increases linearly with the spatial resolution. Again, the fastest algorithms are FAST and AGAST, while ORB, CenSurE and BRISK exhibit slightly worse performance. On the other hand, the processing time of SURF and SIFT is noticeably higher than that of all the other detectors.

2) Repeatability: The detectors are also evaluated with the repeatability metric, defined in [**?**] as the ratio between the number of point-to-point correspondences and the minimum number of keypoints detected in a given pair of images. For the evaluation the dataset provided by Mikolajczyk[2] was used. Each sequence includes a reference image and a set of images that are progressively modified by one or more geometric or photometric transformations. In particular, we selected the following sequences:

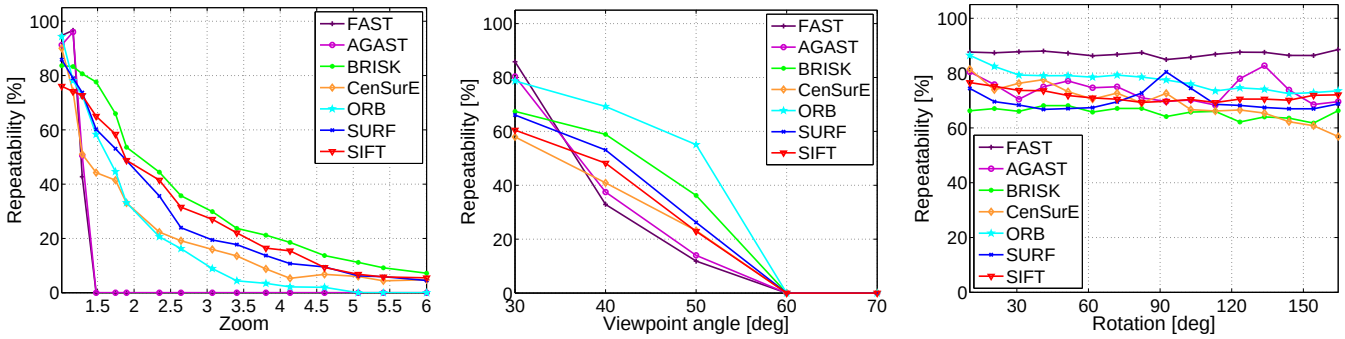
- *Asterix* (17 images with increasing zoom factor);
- *Graffiti* (6 images with increasing view point angle);
- *New York* (17 images with increasing rotation).

To match correspondent regions in the image pairs, ground-truth homography matrices are provided. Following the same approach as in [**?**], two keypoints were considered as matching if: i) their relative position error with respect to the ground-truth is less than 1.5 pixels; ii) their corresponding neighborhoods are overlapped at least by 40%. The repeatability scores for the three considered image sequences are shown in Figures **??**-a, **??**-b and **??**-c.

As expected, the repeatability of non-scale invariant detectors (FAST and AGAST) is very poor in case of zoom transformations (see Fig. **??**-a, *Asterix* sequence). Among the scale invariant detectors, SIFT, BRISK and SURF exhibit the best performance. As for the *Graffiti* sequence (see Fig. **??**-b), the highest repeatability is achieved by the ORB detector; CenSurE, SURF, SIFT and BRISK also obtain good performance, while the repeatability of FAST and AGAST decreases as the view-point angle increases. Moreover, for all detectors, the repeatability drops to zero for angles exceeding 60 degrees. As for rotation transformations (see Fig. **??**-c, *New York* sequence), the repeatability score is almost independent from the angle, and reasonably high (always above 60%). The best score is obtained by the FAST detector, approaching 87% of repeatability on the average; ORB and AGAST also present a very high repeatability score, which is about 77% and 74%, respectively. The average repeatability score of SIFT, SURF, STAR and BRISK are 72%, 70%, 69% and 65%, respectively.

[1] http://www.robots.ox.ac.uk/~vgg/data/oxbuildings/

[2] http://lear.inrialpes.fr/people/mikolajczyk/Database/index.html



(a) *Asterix* sequence. (b) *Graffiti* sequence. (c) *New York* sequence.

Fig. 3. Repeatability score for different image sequences.

III. Descriptor Evaluation

A. Non-binary descriptors

1) SIFT: For each detected keypoint, SIFT assigns an orientation α by selecting the angle that represents the mode of the histogram of local gradients (computed for each pixel around the keypoint). Then, a region of points around the keypoint, oriented as α, is divided into 16 sub-regions, and an orientation histogram with 8 bins is created from the (smoothed) samples for each region. The descriptor is then obtained by concatenating these 16 histograms, leading to a final descriptor of 128 elements in length. The descriptor is finally normalized to unit length to achieve robustness against illumination changes.

2) SURF: For a keypoint detected at (x, y, σ), a circular region centered in (x, y) with size proportional to σ is convolved with two Haar wavelets along orthogonal directions. The results are represented as two dimensional vectors and summed up within a rotating angular window. The longest resulting vector determines the orientation α of the keypoint. Then a square region centered in (x, y), oriented as α and with size proportional to σ, is split up in a grid of 4×4 sub-regions. For each sub-region a 4-dimensional feature vector is defined as:

$$[\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|] \,, \tag{2}$$

where d_x and d_y represent the result of Haar wavelet filtering and the sums are computed over a predefined set of sample points in the respective sub-region. The final descriptor is obtained by concatenating the feature vectors of all the sub-regions, yielding a descriptor vector with 64 elements.

B. Binary descriptors

To build a binary descriptor it is only necessary to compare the intensity between two pixel positions located around the selected keypoints. This allows to obtain a representative description at very low computational cost. Moreover, matching binary descriptor just requires the computation of Hamming distances, which can be executed very fast through XOR primitives on modern architectures.

1) BRIEF: The first work in this area is represented by the Binary Robust Independent Elementary Features (BRIEF) [**?**]. A binary descriptor for a patch of pixels of size $S \times S$ is built by concatenating the results of the following test:

$$b = \begin{cases} 1, & I(\mathbf{p}_j) > I(\mathbf{p}_i) \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where $I(\mathbf{p}_i)$ denotes the (smoothed) pixel intensity value at $\mathbf{p}_i$, and the selection of the location of all the $\mathbf{p}_i$ uniquely defines a set of binary tests. In BRIEF, the sampling points are drawn from a zero-mean isotropic Gaussian distribution with variance equal to $\frac{1}{25}S^2$. To increase the robustness of the descriptor, the patch of pixels is pre-smoothed with a Gaussian kernel with variance equal to 2 and size equal to 9×9 pixels. The authors shows that by using 256 binary tests the BRIEF performance is similar to the SURF performance.

2) ORB: Similar to BRIEF, ORB uses a bi-dimensional Gaussian distribution for generating the sampling pattern. Additionally, it provides rotation invariance to the descriptor, estimating the patch rotation using the intensity centroid, which is computed from the patch moments and is shown to outperform gradient-based approaches. The sampling pattern is than steered with the estimation orientation and the descriptor is built with the usual binary tests. Finally, a subset of binary tests is chosen in order to reduce their inter-correlation, hence increasing the descriptor discriminative power. The algorithm is greedy, and selects the sampling pairs with the highest variance and stops when 256 binary tests are selected.

3) BRISK: The BRISK descriptor uses a pattern of points $\mathbf{p}_i$ equally spaced on concentric circles centered at the keypoint. The pattern defines two sets of point pairs, namely long-distance pairings and short-distance pairings. The long-distance set is composed by all those pairs (i, j) such that $\|\mathbf{p}_i - \mathbf{p}_j\|_2 > \delta_{min}$ and they are used to estimate the orientation of the keypoint by local gradient averaging. Once the keypoint orientation is estimated, Gaussian smoothing is applied, the sampling pattern is rotated and the short-distance pairings (whose pairwise distance is less than a threshold δ_{max}) are used to build the descriptor. In the original implementation, δ_{max} is tuned so that the descriptor has 512 bits.

4) FREAK: Following BRISK, the Fast Retina Keypoint (FREAK) [**?**] uses a pattern of points which is also circular, but with higher density of points near the center. As for rotation invariance, the patch orientation is also estimated by summing local gradients over selected pairs. For the creation of the descriptor, the authors propose an approach similar to ORB by selecting, with a greedy algorithm, the pattern tests which are less correlated and thus more discriminative. For maximum performance, 512 binary tests are used.

C. Experiments

The descriptors introduced in Sections **??** and **??** are now evaluated. All the tests were conducted considering the OpenCV v2.4.4 implementation of the algorithms, except for BRISK, where we considered the original authors' implementation.

1) Processing Time: In this paragraph we show the processing time of the visual descriptor algorithms listed above. For this evaluation, we forced the descriptors to be computed on keypoints of fixed size (64×64 pixels), in order to provide a fair comparison among the different algorithms. Also in this case, the experiments have been carried on a personal computer powered by an Intel i7-3770 CPU at 3.40GHz), with 8GB of RAM. Again, all tests were executed on a single core. Table **??** reports the average processing time for computing 500 descriptors. It is important to observe that the computation of binary descriptors (BRIEF, ORB, BRISK and FREAK) is particularly efficient compared to that of non-binary descriptors. For instance, BRISK is 6 times faster than SURF and 20 times faster than SIFT.

TABLE I
AVERAGE PROCESSING TIME OF VISUAL DESCRIPTOR ALGORITHMS FOR THE COMPUTATION OF 500 DESCRIPTORS.

Descriptor	Proc. time
SIFT	43.45 ms
SURF	13.43 ms
BRIEF	1.43 ms
ORB	1.36 ms
BRISK	2.11 ms
FREAK	1.09 ms

2) Matching accuracy: A comparative analysis was conducted on four ground-truth datasets provided by Winder et. al. [**?**], each containing thousands of patches (64×64 pixels). In particular, the *Liberty* and *Notredame* datasets was considered, with patches corresponding to keypoints computed with both DoG and Harris detectors (refer to [**?**] for details). Using the ground-truth information provided in the datasets, $20k$ pairs of matching patches (i.e., coming from the same 3D point), and $20k$ pairs of non-matching patches were selected for each dataset. For each pair, the distance[3] between the corresponding descriptors was computed and then accumulated into matching and non-matching histograms, respectively. From these histograms we traced the Receiver Operator Characteristic (ROC) curves [**?**], which plot the True Positive (TP) rate (i.e., the percentage of correct matches over the total number of matches) against the False Positive (FP) rate (i.e., the percentage of incorrect matches). The ROC curves relative to the descriptors under evaluation are shown in Figures **??**-a,b,c,d. As shown, the results are rather similar for all datasets. As far as non-binary descriptors are concerned, the SIFT descriptor has superior performance over the SURF descriptor. Among the binary descriptors, ORB and BRISK achieve the best performance, the former being particularly effective for datasets based on the Harris detector (see Figures **??**-b and **??**-d). It is interesting to notice that BRISK results are similar to those of the SURF descriptor, and ORB even outperforms SURF.

[3]Euclidean and Hamming distances were considered for non-binary and binary descriptors, respectively.

IV. IMAGE RETRIEVAL EVALUATION

After the standalone performance evaluation of the visual feature detectors and descriptors, their efficiency for an image retrieval task is evaluated. In this case a real content-based image retrieval system is implemented to search for the most similar images to a given query, in a large database of images.

A. Experimental setup

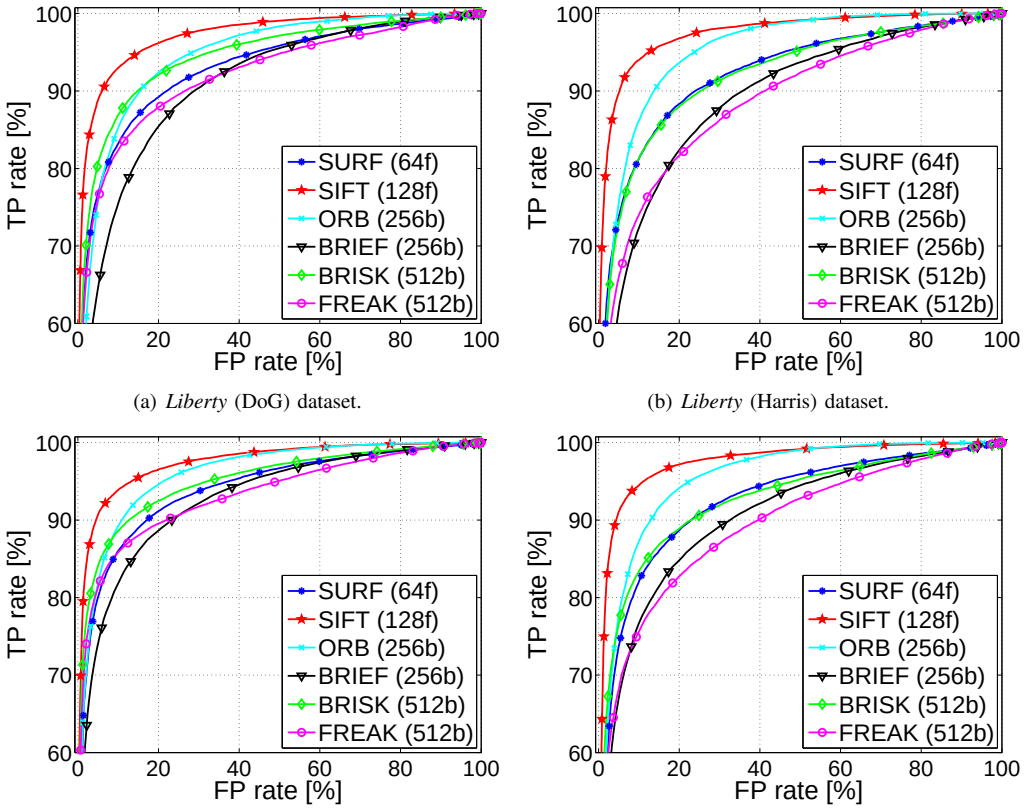
Given the set of K images $I_1, \ldots, I_K$ composing the image database to be queried, each image is represented with a maximum of $N = 500$ visual descriptors. In order to choose the N most relevant descriptors, all the descriptors are sorted according to the keypoint response, as computed by OpenCV. When a query image arrives to the system, a relevancy score is computed for each one of the images. The query descriptors are matched to the descriptors extracted from each one of the images in the database. A descriptor is matched to other if it fulfills a nearest-neighbor distance ratio test with a threshold $\alpha = 0.7$ [**?**]. The number of matched descriptors is taken as the relevancy score of the database image with respect to the query image. The metric employed to measure the quality of the image retrieval system is the Mean Average Precision (MAP). For each query q a rank of N documents in the database is obtained. The Average Precision (AP) for the retrieved list of a query q is defined as

$$AveP(q) = \sum_{k=1}^{N} P(k)\,\Delta R(k)\ , \tag{4}$$

where $P(k)$ denotes the precision of the first k documents and $\Delta R(k) = 1$ if the recall level has changed with respect to $k-1$ and $\Delta R(k) = 0$ otherwise. Intuitevely, the Average Precision represents the area under the precision-recall curve for each query to the system. The MAP is computed by averaging the AP values obtained for a set of test queries $Q = q_1, q_L$

$$MAP = \frac{1}{|Q|} \sum_{q \in Q} AveP(q)\ . \tag{5}$$

To evaluate the performance of the visual descriptors for the image retrieval task, three different datasets are employed:



(a) *Liberty* (DoG) dataset. (b) *Liberty* (Harris) dataset.

(c) *Notredame* (DoG) dataset. (d) *Notredame* (Harris) dataset.

Fig. 4. Receiver Operator Characteristic (ROC) curves for different ground-truth datasets.

- *ZuBuD*: Zurich Building Database[4] contains 1005 images from 201 buildings in Zurich. This dataset provides a test set composed of 115 query images.
- *Oxford*: Oxford Building Database contains images from 16 buildings in Oxford. Only the subsets tagged as good and ok were used. This dataset provides a test set composed of 55 query images.
- *CTurin180*[5]: This database contains 1440 images from 180 buildings in Turin. A test set has been created by extracting from the dataset the 2nd view of the 4th camera for each building to obtain a total of 180 query images. The images have been resized to 800×600 pixels to improve evaluation speed.

With the aim of providing an independent evaluation of descriptors with respect to detectors, different combination pairs have been tested. However, not all combinations are possible, e.g. detectors that are not scale-invariant (e.g. FAST) are not matched to scale invariant descriptors and vice versa. The following combinations are evaluated: 1) corresponding pairs of detector/descriptor as originally proposed in the literature 2) using the SURF detector with all possible descriptors 3) using the ORB detector with all possible descriptors 4) Using the SIFT detector with BRIEF, BRISK and FREAK. The detector and descriptor have the same parameters used for their independent evaluations shown in sections II and **??**. In particular we considered the OpenCV v2.4.4 implementation of all the algorithms, except for BRISK, where the original implementation was adopted for both the detector and the descriptor.

[4]http://www.vision.ee.ethz.ch/showroom/zubud/index.en.html
[5]http://pacific.tilab.com/wordpress/?page_id=5

B. Results

Figure **??** presents the MAP values obtained for the three datasets and the different combinations of detectors and descriptor evaluated. From the experimental results obtained, it is possible to conclude that:

- Comparing the SURF with ORB and SIFT detectors, the best MAP results are obtained for the SURF detector in the *Oxford* and *CTurin180* datasets and for most combinations in the *ZuBuD* dataset.
- In average, the BRISK descriptor gives the best MAP results independently of the employed detector. This can be observed for the *ZuBuD* and *CTurin180* datasets where BRISK is the best descriptor and for the *Oxford* dataset where SURF-BRISK is the second best.
- Overall, the best MAP results are obtained when the SURF detector is combined with the BRISK descriptor by averaging over all datasets.
- MAP values are much lower for the *Oxford* dataset than for *ZuBuD* and *CTurin180* datasets. This occurs because

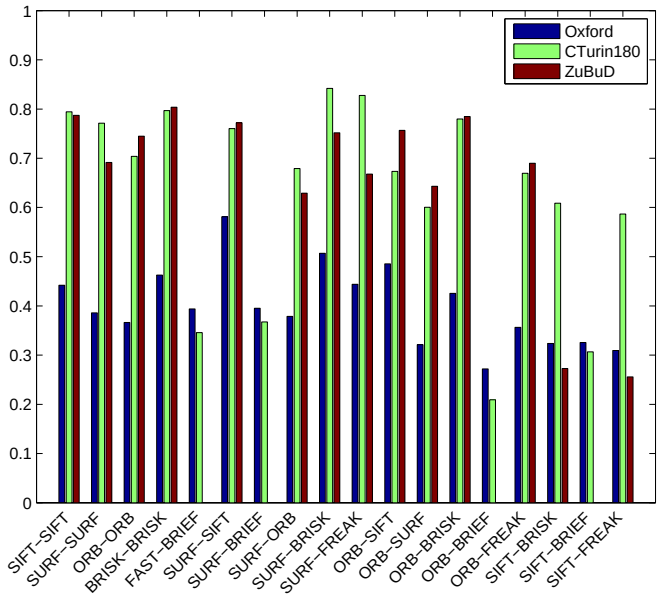


Fig. 5. Mean Average Precision values for the *ZuBuD*, *Oxford* and *CTurin180* datasets with diffferent combinations of detectors and descriptors.

the number of relevant images for each one of the queries in the *Oxford* dataset is much higher. The evaluation results also show that, in the design of the image retrieval system, the selection of the descriptor, jointly with the employed detector, is rather important. For example, for the *CTurin180* dataset, the SURF detector can bring MAP improvements of up to 10% with respect to the BRISK-BRISK combination.

V. Conclusions

This paper has presented a comprehensive evaluation of visual feature detectors and descriptors, which were evaluated both standalone and for an image retrieval task. For the independent evaluation of the detectors, the processing time and repeatability score were calculated while for the descriptors, matching accuracy was assessed. For the image retrieval, a joint evaluation of the detectors and descriptor was done with the MAP relevance metric. The results obtained show that a good score in the standalone evaluation did not always lead to high accuracy in the image retrieval task. The evaluation has shown that binary descriptors can achieve a performance similar to (and sometimes higher than) to that of non binary descriptors, with much lower complexity. As future work, this evaluation should be performed for other scenarios where binary descriptors can bring benefits, such as object recognition, 3D reconstruction and wide baseline stereo.