BOLD - Binary Online Learned Descriptor For Efficient Image Matching

Vassileios Balntas Lilian Tang Krystian Mikolajczyk
University of Surrey, UK
{v.balntas,h.tang,k.mikolajczyk}@surrey.ac.uk

Abstract

In this paper we propose a novel approach to generate a binary descriptor optimized for each image patch independently. The approach is inspired by the linear discriminant embedding that simultaneously increases inter and decreases intra class distances. A set of discriminative and uncorrelated binary tests is established from all possible tests in an offline training process. The patch adapted descriptors are then efficiently built online from a subset of tests which lead to lower intra class distances thus a more robust descriptor. A patch descriptor consists of two binary strings where one represents the results of the tests and the other indicates the subset of the patch-related robust tests that are used for calculating a masked Hamming distance. Our experiments on three different benchmarks demonstrate improvements in matching performance, and illustrate that per-patch optimization outperforms global optimization.

1. Introduction

Significant progress has been made in creating new feature descriptors that are either based on floating point arithmetic, such as SIFT [9], SURF [1] and GLOH [11] or on binary strings and hamming distances like BRIEF [3], ORB [14] and BRISK [8].

Large datasets with correspondence ground truth enabled learning methods to be used to improve the descriptor performance [19]. One such approach consists of optimally learning descriptor parameters [20]. Another research direction is learning discriminative projections from high dimensional feature space to subspaces with better determinability. In [10, 2] the descriptor optimization is similar to the LDA based projections, which simultaneously minimizes distances intra-class and maximizes them inter-class. Similarly, the authors of [15] propose a convex optimization for descriptor learning. However, in all these methods, the intra-class is formed by positive examples of correctly matched patch pairs while in LDA by various instances of the same image category / content. LDA projections cannot be learned for each patch independently due to practical complexity issues, *e.g.* inefficient distance calculation and matching. Thus in the case of image patches, discriminant projections are learned globally and lead to a limited improvement. Local discriminant projections are expected to give better results if adapted to each class independently.

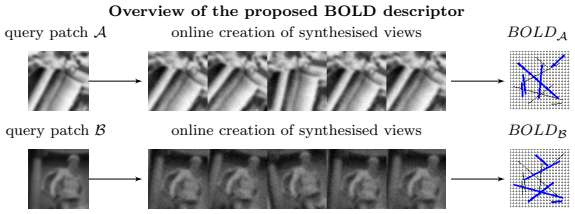

Figure 1. In contrast to typical approaches that use the same measurements for all patches, we adapt the descriptor online to each patch. The blue line ends indicate the selected binary tests from a common superset, based on the measurements from the synthesized views of each patch.

In the context of binary descriptors, BRIEF was improved in [14] by selecting uncorrelated tests that maximize the variance across training patches. Learning of discriminant and low dimensional spaces has also been applied to binary descriptors. DBRIEF [17] is built by using the inter to intra class distance objective adapted to a binary descriptor. A set of discriminative projections is computed and approximated with a set of predefined dictionaries in order to generate a binary feature vector. The recently proposed BinBoost descriptor [16] applies boosting to learn a set of binary hash functions that achieve a performance comparable to real-valued descriptors. Both DBRIEF and BinBoost are not based on binary tests therefore the extraction process is less efficient. A different research direction is to use coding methods to make the descriptors more compact [4].

The various feature descriptors proposed in the literature differ in design, theory and implementation, but a common approach is the computation of the final feature vector from a fixed set of measurements that are applied to all described patches. It follows that the measurements are not varied depending on the content of the patch. This is based on

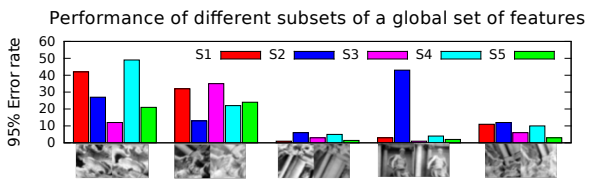

Figure 2. Error rates across different sets of binary tests `Sx` for 5 classes of the Trevi dataset [19]. The performance of each set significantly varies across different patch classes.

important practical considerations which primarily include convenience in using various distance metrics and efficient matching techniques for large scale problems. Moreover, learning based components are trained offline as they are typically too computationally intensive for any online processing. In BRIEF descriptor [3], four different arbitrarily designed configurations of binary tests were evaluated on an entire training set and the best performing configuration was selected. However, intuitively different patch appearances can be best represented by different measurements. For example, the results from [18] show that recognition performance can be improved by adapting the spatial structure of SIFT-based descriptors to each class.

In this paper we propose an approach which combines the advantages of efficient binary descriptors with the improved performance of learning-based descriptors. We demonstrate that there is no single set of measurements that is globally optimal for all patches in a dataset and significant improvement can be gained by adapting the binary tests to the content of each patch. The measurements are first designed to maximize the inter-class distances and then a subset is selected online for each patch to minimize the intra-class distances. This is done efficiently in such a way that the extraction time is comparable to other binary descriptors. The proposed online selection of discriminative binary tests can be applied to other techniques such as decision trees or ferns. Nearest neighbour matching of descriptors is also efficient by calculating a modified Hamming distance. We evaluate the proposed descriptor on different benchmarks and demonstrate performance that matches that of SIFT, with computational efficiency that matches that of BRIEF.

2. Intra and inter-class optimization of binary descriptors

In this section we first demonstrate the improvements in matching accuracy that can be obtained by adapting a set of binary tests to the input. We then present a method for adaptive discriminative selection of binary tests, and its efficient implementation.

2.1. Performance of random binary tests

In descriptors such as BRIEF where random tests are involved, a random number generator is typically used with an arbitrary seed which guarantees the repeatability of generated values. For example the OpenCV implementation of BRIEF uses 42 as the seed. To demonstrate how the performance can be affected by using different feature sets we generate 5 different binary test sets that are subsets of a larger set based on five different random seeds. All these sets are generated using the Gaussian distribution proposed in [3], therefore with a large number of tests the final descriptors appear very similar to each other. However, their ability to robustly represent different image patches varies significantly. This is illustrated in Figure 2 where 5 classes from the Trevi dataset [19] are shown, together with the 95% matching error rates for each of the feature sets. The error rate is the percentage of false matches when 95% of correct matches are obtained, which was used in other evaluations on this dataset [20, 16]. Each class consists of 5-8 patches that originate from the same 3D point and the distances between these patches are referred to as intra-class distances.

The first observation from Figure 2 is that the matching error for some patches is significantly higher than for others independently of the feature set used. However, another important observation is that although the feature sets differ only in the exact random locations of their binary tests, their performance significantly varies for different patches. *E.g.* `S3` is the best performing descriptor for the first class, while it is the worst for the second class. `S2` gives an error rate larger than 40% for the 4^{th} class, while all the other descriptors give error rates very close to 0%.

Typically, a global optimization such as the one used in DAISY [20] or BinBoost [16], aims at finding a configuration that leads to the lowest average error across all classes in a dataset. In our example `S5` is globally the best with the average error rate of approximately 11%. At the same time, by choosing the best performer for each class independently, an overall error rate of 5% could be achieved. Note that the configuration `S3` uses the same seed as the `openCV` library. From the results above, it is clear that a method that can locally adapt the descriptor extraction method to each patch will outperform a global fixed configuration, regardless of the size of the global training data.

A naive implementation of this idea is to store a version of the entire dataset for each locally adapted descriptor. However, the practical implications such as memory and computational complexity in particular, are difficult to deal with. Another issue is that while in the experiment above we can identify the best configurations per patch a posteriori, a method to identify such descriptors a priori using as only input the patch to be described is not obvious.

2.2. Learning discriminative descriptors

It has been frequently demonstrated that descriptors perform better when the separation between the intra-class distances and the inter-class distances is maximized. Given a set of labelled matching and non-matching image patches, methods like [2, 10] seek to find a projection $\mathbf{w}^*$ s.t. $\mathbf{w}^* = \arg\max_{\mathbf{w}} (\mathbf{w}^T\mathbf{A}\mathbf{w})/(\mathbf{w}^T\mathbf{B}\mathbf{w})$ which is the ratio of the inter $\mathbf{A}$ to intra-class $\mathbf{B}$ covariance along the direction $\mathbf{w}$. Intuitively, such methods seek to minimize the expected distance between patches annotated as similar and maximize the expected distance between patches annotated as dissimilar. This has been done globally for real-valued descriptors in [2, 10, 17] with the use of a large set of negative and positive pairs of patches in an offline learning process.

In the following we propose an approach that exploits this idea to optimize a binary descriptor for each patch independently.

2.3. Properties of binary tests

Let $\{\mathbf{x}_1, \mathbf{x}_2, \cdots \mathbf{x}_N\}$ denote a set of binary descriptors of dimensionality D, extracted from N patches which can be arranged in matrix $\mathbf{X}$ of size $N \times D$. Each column c_i with $i \in [1, ...D]$ represents a test/dimension of the binary descriptors and can be viewed as a binary string of length N that follows a Bernoulli distribution with a certain probability of values 1 or 0. Matrix $\mathbf{X}$ can then be expressed as the outcome of N trials of D Bernoulli distributions $\mathcal{B}_i$ with $i \in [1 \cdots D]$. If the mean value of $\mathcal{B}_i$ is ρ_i, then the variance is $\sigma_i = \rho_i(1-\rho_i)$ where ρ_i is the ratio of `1s` and $(1-\rho_i)$ is the ratio of `0s` in column c_i. Variance σ_i of the i^{th} dimension has a direct relation with the Shannon entropy of the binary string of the corresponding column c_i i.e. $\mathcal{E}_i = -\rho_i \cdot log_2\rho_i - (1-\rho_i)log_2(1-\rho_i)$.

A required characteristic of such binary strings is to exhibit a high variance–entropy values if descriptors $\mathbf{x}_n$ belong to different classes and a low variance–entropy values if descriptors $\mathbf{x}_n$ belong to the same class. For the former, the discriminative dimensions are the ones where the variance reaches the maximum possible value of 0.25 (entropy reaches 1). The latter implies that the process that generates the values for this specific descriptor dimension, is stable and robust to noise, deformations, illumination changes etc. In an ideal case, with a perfect descriptor all columns of intra class descriptors $\mathbf{X}$ would have entropy and variance equal to zero. Given $\mathbf{X}$ and Bernoulli distributions $\mathcal{B}_i(\rho_i, \sigma_i)$ associated with test/dimension i of $\mathbf{X}$, the expected average distance $\mathbb{E}[\Delta]$ between descriptors in $\mathbf{X}$ is related to the sum of the variances σ_i. This can be derived from:

$$\mathbb{E}[\Delta_{intra}] = \frac{1}{D}\sum_{i=1}^{D}\mathbb{E}[\Delta_i] \tag{1}$$

where $\mathbb{E}[\Delta_i]$ is the expected intra-class distance value for dimension i:

$$\mathbb{E}[\Delta_i] = \frac{1}{N^2}\sum_{m=1}^{N}\sum_{n=1}^{N}|x_{m,i} - x_{n,i}|_{\oplus} \tag{2}$$

where $|x_{m,i} - x_{n,j}|_{\oplus}$ is the Hamming distance between two binary values. Since $|x_{m,i} - x_{n,j}|_{\oplus} = (x_{m,i} - x_{n,j})^2$ we obtain:

$$\begin{aligned}\mathbb{E}[\Delta_i] = \frac{1}{N^2}\sum_{m=1}^{N}\sum_{n=1}^{N}x_{m,i}^2 - 2\frac{1}{N^2}\sum_{m=1}^{N}\sum_{n=1}^{N}x_{m,i}x_{n,i} \\ +\frac{1}{N^2}\sum_{m=1}^{N}\sum_{n=1}^{N}x_{n,i}^2 = 2\mathbb{E}[x_i^2] - 2\mathbb{E}[x_i]^2\end{aligned} \tag{3}$$

The variance of dimension i is therefore directly reflected by the fraction of 1s in column i of matrix $\mathbf{X}$. From the above it is clear that dimensions with high variance increase the intra-class distances, and dimensions with low variance decrease it. Low variance is required for descriptors from the same class (positive patches) and high variance for descriptors from different classes (negative patches).

It was demonstrated in [2] that discriminant projection of SIFT dimensions be achieved in a two stage process which first diagonalizes the intra-class covariance and then performs a global PCA. Thus the dimensions are decorrelated and oriented along dominant directions in the real-valued space. This process can be adapted to learning of discriminative binary descriptors by first selecting uncorrelated the tests/dimensions that maximize the inter-class distances globally and then by short-listing tests that minimize the intra-class distances locally. Correlation $\mathbb{C}_{ij}$ between tests i and j in matrix $\mathbf{X}$ can be measured on inter-class patches by the Hamming distance between the corresponding columns:

$$\mathbb{C}_{ij} = |\frac{2}{N}\sum_{m=1}^{N}|x_{m,i} - x_{m,j}|_{\oplus} - 1| \tag{4}$$

Thus the value of $\mathbb{C}_{ij}$ varies between 0 and 1, with 1 for perfectly correlated tests. Suitable dimensions can be chosen by thresholding this measure.

The first two steps of the process, the global selection of discriminative dimensions and the decorrelation can be done offline from a large set of possible binary tests and random patches. The final selection of tests that minimize the intra-class variance has to be done per patch which requires efficient online implementation.

2.4. Efficient extraction of online learned descriptors

In this section we present the details of our online learned descriptor. As discussed in the previous section this is done

in two steps, namely inter-class offline optimization and intra-class online selection of tests.

2.4.1 Global optimization

Global optimization is based on a large set of N diverse image patches of normalized size from the `Trevi` dataset [19] which is different from the datasets used in our experiments.

Our features $f_i = (t_1, t_2)_i$ are sets of binary tests that consist of comparing pixel intensities in pairs of locations t_1 and t_2 within the patch. For a grid of $P \times P$ locations within a patch (*e.g.* $P = 32$) the total number of tests is $M = \binom{P^2}{2}$. Further constraints on how test are generated can be introduced here. These may exclude locations on patch boundaries, large distances between t_1 and t_2, etc.

In global optimization the goal is to identify the subset of discriminative features. In the case of binary tests, this consists of finding features that give a large variance across inter-class examples as discussed in section 2.3. This requires calculation of all test responses in each of the N patches. It results in a set of N binary strings of dimensionality M with $\mathbf{x}_n$ representing the bitstring of patch n. $\mathbf{X}$ is a matrix with descriptors $\mathbf{x}_n$ as rows. We then calculate the fraction of 1s in column i of $\mathbf{X}$ and sort the columns according to that measure. This ranks high the discriminative tests, which exhibit a high variance across a random set of inputs.

The next step is to select a subset of uncorrelated features. We follow the greedy approach from [14] which starts by selecting the first high variance tests from the ranked list and then searches for another high variance test with the correlation score $\mathbb{C}_{ij} < \tau_C$ (*e.g.* $\tau_C = 0.2$). The process continues by verifying at each iteration the correlation between the candidate and all selected tests. The selection stops when a defined number G of tests has been found (*e.g.* $G = 512$).

Note that the global optimization is done offline as it concerns to whole set of possible tests and diverse image patches that represent negative examples in section 2.3.

2.4.2 Local online learning

As demonstrated in [16, 14] a set of globally optimized tests outperform a set of random tests in terms of matching error rates. However, as we show in Figure 2 different subsets of tests minimize the intra-class distances for individual classes of patches and can achieve superior performance compared to the globally optimized features.

To fully benefit from the LDA-like optimization, intra-class distances have to be minimized. We consider each patch as a separate class, therefore this optimization has to be performed online during descriptor extraction. Given that a patch is a single instance from a class, additional examples have to be synthetically generated to estimate intra-class variance $\mathbb{E}[\Delta_i]$. This approach proved successful in many applications, in particular in the context of local image patches affine projections are typically applied [2, 13].

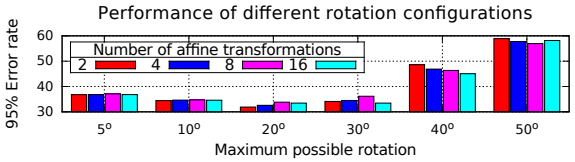

Figure 3. 95% matching error rate for Yosemite 100k with respect to various affine examples in intra-class optimization of binary tests. Small affine transformations and few examples are sufficient to achieve low error rate.

Generating various geometric views of the same patch can be done easily (*e.g.* with affine matrices and bilinear interpolation), but in large datasets or real time applications the computational complexity would grow significantly. However, given the globally optimized set of binary tests, which is of a limited size, we can avoid the need for patch warping by applying the geometric transformations directly to the pixel locations $(t_1, t_2)_i$ of each test f_i rather than to the image patches. For each feature, a new set of features can be created, which consists of affine-transformed versions of itself. Furthermore, since the set of tests is fixed, the locations of tests under various affine transformations can be stored in a lookup table rather than calculated online. Thus, our set of tests is extended in a lookup table to different $f_{ia} = (t_1, t_2)_{ia}$ where a indicates an affine transformation of test f_i.

We examined various affine transformations to generate intra-class variances and to identify stable tests. We report the results in Figure 3 in terms of 95% error rate for 100k patches from the `Yosemite` dataset. Parameters of affine transformations to generate positive examples were extensively studied in [2] with the conclusion that small random transformations lead to better results. We make similar observations and notice that small affine projections with a maximum rotations of 10^o to 20^o are the ones that give the best results. It is also worth noting that as few as 2 transformations are sufficient to identify tests that minimize the intra-class variance. This is an important observation as a small number of transformations leads to few affine lookup tables that need to be created. This then leads to more efficient online evaluation of binary tests which consist only of sampling and comparing pixel values in the tests.

Given the binary strings generated by tests f_{ia} and represented in intra-class matrix $\mathbb{X}_{ia}$, a subset of tests f_i that minimizes the variance along dimension a is selected. In our implementation we select only the tests for which the variance is 0. However more complex methods can be applied, such as variance sorting and thresholding.

Having identified the sets that are to be included in the

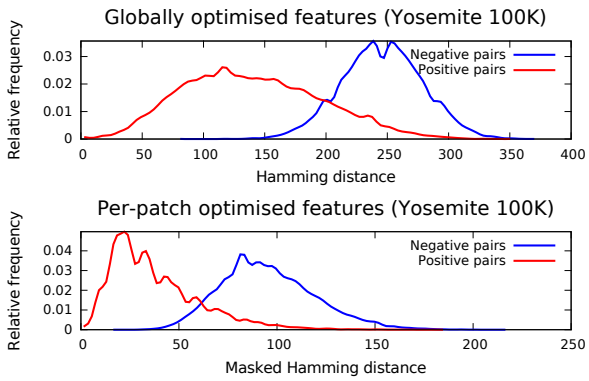

Figure 4. Negative (inter-class) and positive (intra-class) distance distribution of globally (top) and locally (bottom) optimized descriptors. The intersection area between the two distributions is reduced from 13.97% to 9.75% for globally vs. locally optimized descriptors. Thus the number of mismatches is reduced.

BOLD descriptor, each patch is represented by the results $\mathbf{x}_n$ of the adapted binary tests and a second binary string $\mathbf{y}_n$ of length G where 1s indicate which tests are valid for patch n. D_n is the number of 1s in $\mathbf{y}_n$ which may differ for every patch n.

2.4.3 Matching of locally adapted descriptors

After global and local online selection of discriminative tests during descriptor calculation each patch is represented by a binary string $\mathbf{x}_n$ and a binary mask $\mathbf{y}_n$, both consist of G bits. The matching of two descriptors is done using the following symmetric Hamming distance between the descriptors and their masks:

$$\mathbb{H}(\mathbf{x}_m, \mathbf{x}_n) = \frac{1}{D_m}\mathbf{y}_m \wedge \mathbf{x}_m \oplus \mathbf{x}_n + \frac{1}{D_n}\mathbf{y}_n \wedge \mathbf{x}_m \oplus \mathbf{x}_n \quad (5)$$

The operation $\mathbf{x}_m \oplus \mathbf{x}_n$ is performed only once and logical AND is performed between the resulting string and the masks $\mathbf{y}_n$ and $\mathbf{y}_m$.

2.4.4 Global vs. local optimization of binary tests

In this section we investigate the properties of the proposed descriptor.

Figure 4 (top) shows the distribution of intra and inter class distances for 512 globally optimized tests. Positive patch pairs from Yosemite dataset represent intra-class and negative pairs correspond to inter-class. The selected tests exhibit high variance across negative patch pairs and small correlation $\mathbb{C}_{ij}$ between tests (*e.g.* < 0.2). In contrast, Figure 4 (bottom) shows distance distributions for our locally optimized tests, where is each patch was described by a different subset of tests from the globally optimized set.

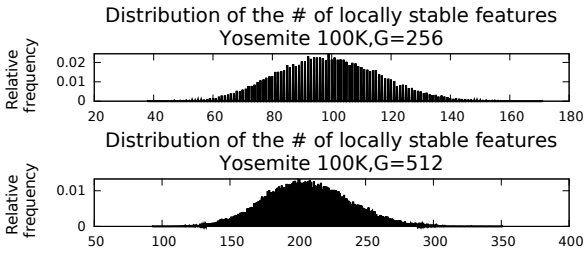

Figure 5. Histograms of descriptor dimensionality after the online selection of locally optimized tests. G denotes the dimensionality of the globally optimized feature set.

The intersection between the distributions for globally optimized tests is 13.95% and for patch adapted ones is 9.75% which corresponds to 30% of relative improvement.

Figure 5 shows the histograms of descriptor dimensionality after online selection of stable tests. G denotes the number of globally optimized tests. We can observe that for $G = 256$ the average number of locally stable tests is ≈ 100 and for $G = 512$ it is ≈ 200, which is approximately half of G. This shows that for each patch, only approximately half of the binary tests are robust to simple affine deformations.

3. Experiments

In this section we present the evaluation results on several datasets and comparisons to state-of-the-art descriptors. For SURF, BRIEF, BinBoost, and DBRIEF, the original implementations provided by the authors were used. For ORB, we use the set of 256 binary tests that are included in OpenCV. For SIFT, we use the implementation from VLFeat.

3.1. Patch dataset

We first evaluate the proposed descriptor using the dataset from [6] and the evaluation protocol from [6, 16], based on ROC curves and error rates. We use a set of $100k$ patches for our experiments, which are resized to 32×32.

In Figure 6 (top) we plot the ROC curves for the full set of the globally optimized binary features of 512 bits compared to the per-patch optimized subsets of our proposed BOLD descriptor. Our method outperforms the global set of features for all false positive rates. This is significant, since it shows the clear advantage of per-patch optimizations compared to global per-dataset optimizations. It has to be noted, that although the final BOLD descriptor has significantly less dimensions involved in the computation of the distances and it is always a strict subset of the globally optimized tests, it outperforms the parent superset of feature dimensions.

In Figure 6 (bottom), we present the results of the comparison between our descriptor and other widely used descriptors such as BinBoost, SIFT, SURF, ORB, DBRIEF,

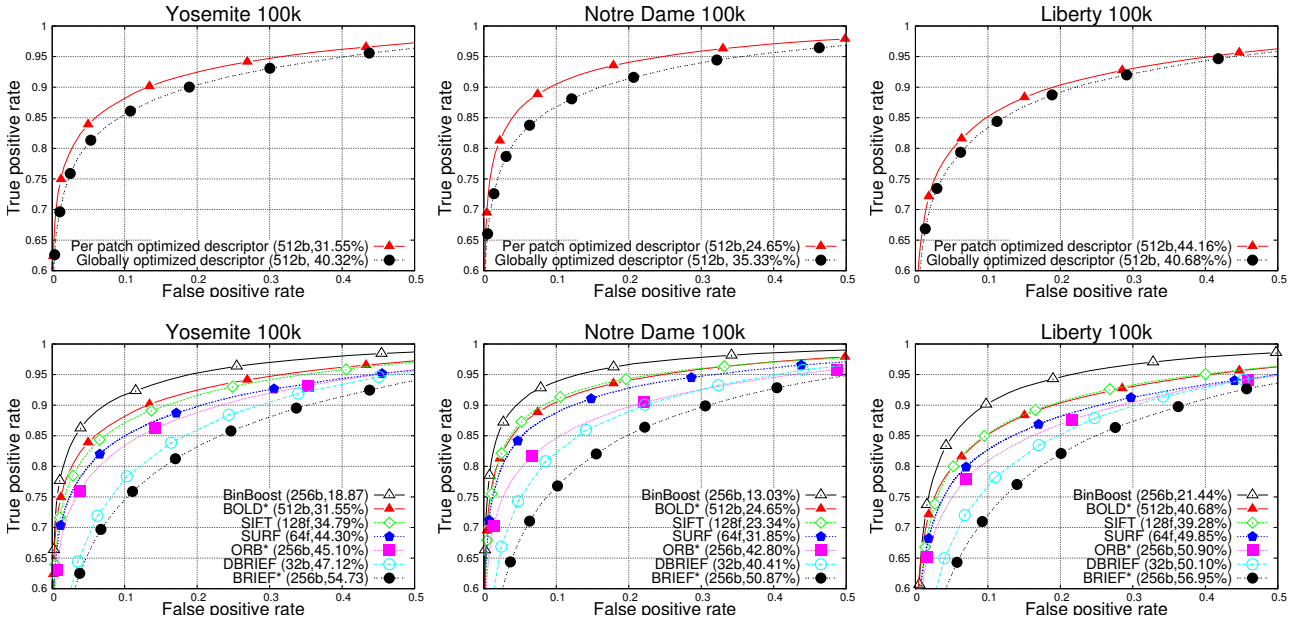

Figure 6. Top: Globally vs. locally optimized features. Bottom: BOLD compared to several state of the art descriptors. Descriptors with * are based on simple intensity tests. Using our per-patch optimization framework, performance of SIFT can be matched with simple intensity tests instead of gradient statistics.

and BRIEF. It is important to note that out of the best performing descriptors i.e. BinBoost, SIFT and BOLD, our descriptor is the only one to use simple binary intensity tests. Both SIFT and BinBoost use quantized gradient responses which are capture significantly more information about the patch statistics. Recently, in [16] it was shown that intensity binary tests are less effective as descriptor dimensions compared to features based on quantized gradients when optimized globally with the same theoretical framework. Our results show however that their performance can be greatly improved by simply using our online per-patch adaptation framework.

The results of the BOLD descriptor compared directly with the other descriptors that are based on simple intensity tests such as BRIEF and ORB, indicate the great performance boost from the proposed method.

3.2. Keypoint matching

In this section, we evaluate the proposed descriptor in image matching, following the framework proposed in [11]. Using the Haris-Laplace detector [12], we extract a set of keypoints from each of the images and normalise them under a canonical representation. We extract a set of descriptors from all those patches and evaluate them with the original protocol from [11]. The results are reported in terms of recall vs. 1-precision, which is computed based on different matching thresholds.

In Figure 7 (top) we plot the results for a pair of images from each sequence from [11] that represents a significant transformation. Results of other image pairs are consistent. Interestingly, SIFT gives the best results overall. However, BOLD outperforms SIFT for high precision part of the curves in Boat, Bikes and Bark sequences. It is worth noting that although BinBoost performs well in the patch dataset, it is ranked third in the matching experiment behind SIFT and BOLD. This may be due to a different training data used to optimize BinBoost and different feature points.

In Figure 7 (bottom) we can also observe the improvement introduced by online selection of binary tests in the intra-class optimization. This advantage of per-patch vs. global optimization is significant and consistently observed in all our experiments on different datasets.

3.3. Tracking by detection

In this section, we demonstrate the application of our method to the tracking by detection problem. Several works [7, 5] follow the tracking by detection approach in which a model is initialized in the first frame, and updated online in order to account for appearance changes. For our experiment, we used the tracking-by-detection mechanism from [7] where the online learned detector is based on random ferns [13].

We build a detector that is trained in the first frame but it is not updated online to avoid the influence of various training examples that can be collected online and alleviate

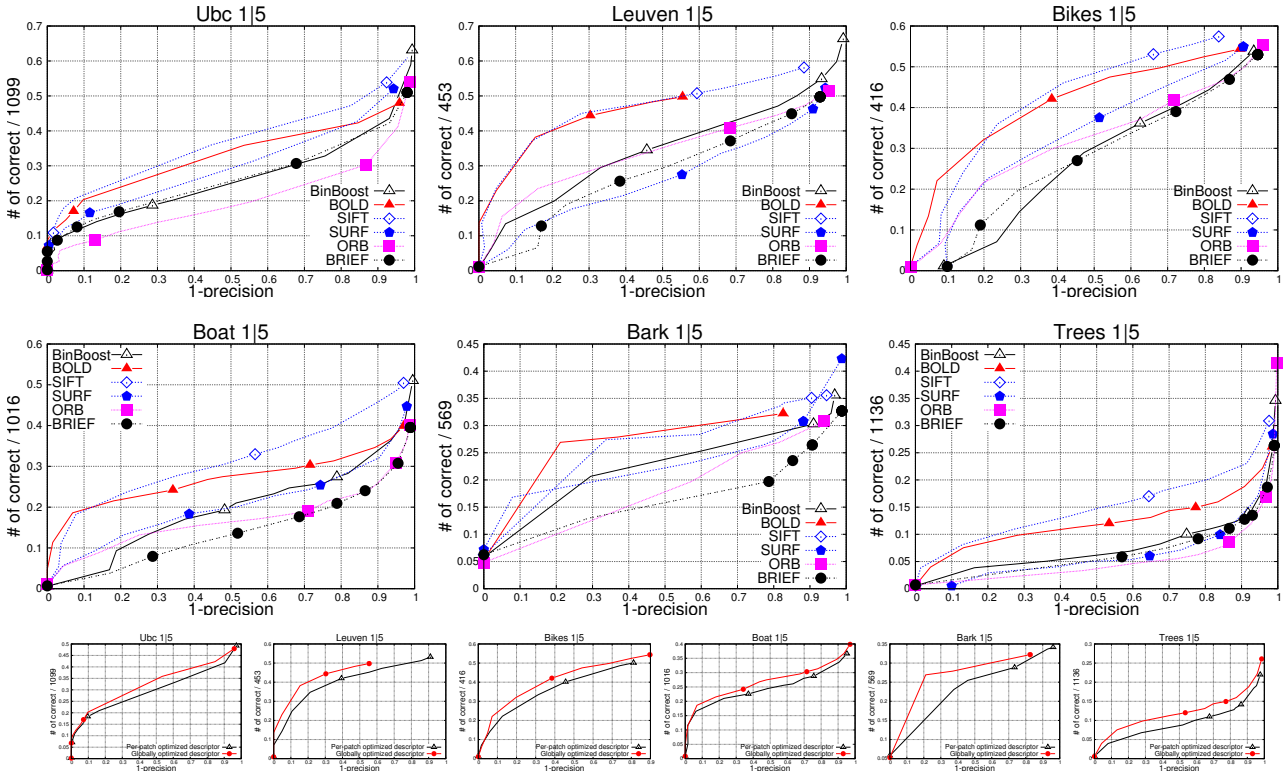

Figure 7. Keypoint matching experiment of the benchmark from [11].

the problem of weak binary tests. Our goal is to show the impact our optimization of the binary tests adapted to the object to be tracked may have in ferns.

Similarly to [13] our goal is to create a classification system based on a set of N simple binary features of intensity differences, similar to the ones in BRIEF and ORB. Following a sliding window approach, which is common among the state of the art detectors, our goal is to classify each window candidate as object or background.

Since each of the f_i features is a simple test, a number of those is required to achieve good detection performance. The authors of [13] apply $\approx$ 300 while the fern classifier in [7] uses $\approx$ 130. A complete representation of the posterior probabilities for each of the background and object classes is therefore impractical due to the large number of used binary tests. Thus in [13] N features are devided into M groups of size $\frac{N}{M}$. Each of those groups forms a fern. The conditional probability becomes $P(f_1, f_2, ...f_N|object) = \prod_{i=1}^{n} P(F_k|object)$. Following [7], we use a sum of the probabilities and a threshold $t_{object} = 0.5$. Thus, if $\sum_{i=1}^{n} P(F_k|object) \geq t_{object}$ we consider it a valid detection.

The goal of this experiment is to demonstrate that the performance of the fern detector depends on the choice of the tests f_i. Full randomization in all stages is proposed in [13], but based on our results from matching the descriptors, we investigate if the per-object adaptation of the binary features that are included in the ferns, can have an effect on the final result.

For the results in Table 1, we use the same detector configuration as in [7] with 10 ferns, each consisting of 13 binary intensity tests. The posterior $P(F_k|object)$ for each fern is learned only from the first frame, using a set of 200 affine transformations of the original patch plus noise.

We generate a pool of 20 ferns, and compare two strategies for the selection of the final 10 that will act as the classifier, one global and one adapted per object. In the first case, we follow the approach of [13] and [7] of randomly selecting a subset. For the second approach, we evaluate the posteriors of each fern in our set of 200 positive examples generated from the object, and we choose the 10 ferns that minimize the intra-class Hamming binary distance across the synthesized 200 positive examples.

We test this method in 10 sequences from the recently published tracking benchmark [21]. We report the recall, which is $\frac{\#\ of\ correct\ detections}{\#\ frames}$. We do not report the precision, since this simple detector/tracker does not update its model online, its precision is therefore 1 or very close to 1 in most cases.

Sequence	Global Ferns	Ferns adapted per object
Subway	0.19	0.28
Jumping	0.26	0.46
Girl	0.44	0.58
Suv	0.25	0.42
Woman	0	0.1
Freeman1	0.07	0.13
Freeman4	0.09	0.16
Deer	0.04	0.18
Crossing	0.3	0.45
Couple	0.03	0.1
Average	0.17	0.29

Table 1. Recall results for 10 sequences of the recently published tracking evaluation benchmark [21]. We observe that selecting a subset of ferns per object outperforms a global set of ferns fixed for all objects.

Distance (512 dimensions)	μS
$\mathbf{x}_L \oplus \mathbf{x}_R$	220
$(\mathbf{x}_L \oplus \mathbf{x}_R \wedge \mathbf{y}_L) + (\mathbf{x}_L \oplus \mathbf{x}_R \wedge \mathbf{y}_R)$	340

Table 2. Performance of the masked Hamming distance, for 1000 pairs of patches. Our proposed masked Hamming distance presents similar efficiency to the original Hamming distance.

The results reported in Table 1 compare the randomly generated tests to object-adapted ferns based on our approach. The per-object optimized ferns perform significantly better than the random tests. Similarly to per-patch online adaptation of descriptors, per-object adaptation of ferns improves the recall of the detectors. Object tracking by detection is an excellent application for the proposed method, since due to the efficiency requirements the learning has to be done online and powerful machine learning methods that require large set of training examples have limited use.

3.4. Timings

In this section, we discuss the computational efficiency of our BOLD descriptor with the proposed masked Hamming distance (cf. Section 2.4.3). The results are averaged on a set of $100k$ patches from the Liberty dataset. All the experiments were done on an Intel i7-Haswell processor with the avx-2 instruction set enabled, and all the possible SIDM optimizations were used (*i.e.* `popcount`).

In Table 2, we compare the calculation time to the regular Hamming distance when matching two binary descriptors. We see that despite the introduction of the symmetric masked Hamming distance, the matching computational efficiency remains very high and comparable to that of the normal Hamming distance, since the only additional operation is the logical AND with the masks.

In Table 3, we report the running times for extraction and matching for several of the descriptors reported in the

Descriptor	extraction	matching	total
BinBoost	713	0.11	713.11
SIFT	417	10	427
SURF	48.2	5	53.2
BOLD	10.5	0.34	10.84
DBRIEF	6.8	0.02	6.82
ORB	2.7	0.11	2.88
BRIEF	2.7	0.11	2.88

Table 3. Comparison of efficiency per operation for various feature descriptors. Time is reported in μS per descriptor.

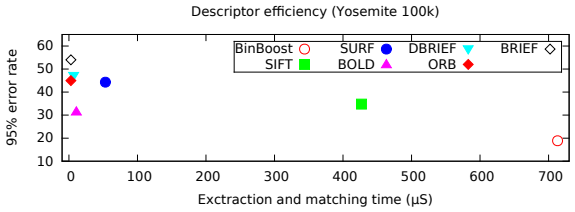

Figure 8. The proposed BOLD descriptor has good properties of of low error rates and high computational efficiency.

results. We can observe that BOLD presents much better results in terms of 95% error rate and remains competitive with BRIEF in terms of both extraction and matching speed.

Furthermore, in Figure 8 we plot the performance of each descriptor in comparison with its computational requirements. We can see that with the proposed framework, we can achieve error rates similar to the SIFT descriptor, with extraction times on the level of BRIEF descriptor.

4. Conclusion

We have proposed a novel approach for generating descriptors that are adapted independently per-patch. Our method relies on binary tests that can be efficiently extracted, evaluated and selected. We present a full inter- and intra-class optimization of binary descriptors that is performed online for each image patch.

The results from several experiments on different datasets show that using a local optimization leads to significant improvements over a global one. Furthermore, the efficiency of the proposed implementation is comparable to other binary descriptors and significantly better than real-valued descriptors. Our approach is the first attempt to use per-patch descriptor with successful results in terms of matching performance and speed in typical computer vision applications.

The proposed method can be applied to other techniques such as decision trees or ferns. An interesting extension would be to apply the proposed selection approach to quantized gradient based features as in BinBoost or SIFT descriptors.

A free and open source implementation of the BOLD descriptor is available at http://vbalnt.io/projects/bold/.

Acknowledgement This work has been supported by EU Chist-Era EPSRC EP/K01904X/1 Visual Sense project.

References

[1] H. Bay, T. Tuytelaars, and L. V. Gool. Surf: Speeded up robust features. In *ECCV*, 2006. 1

[2] H. Cai, K. Mikolajczyk, and J. Matas. Learning linear discriminant projections for dimensionality reduction of image descriptors. *IEEE TPAMI*, 33(2):338–352, 2010. 1, 3, 4

[3] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. Brief: binary robust independent elementary features. In *ECCV*, 2010. 1, 2

[4] V. Chandrasekhar, G. Takacs, D. Chen, S. Tsai, R. Grzeszczuk, B. Girod. CHoG: Compressed histogram of gradients a low bit-rate feature descriptor. In *CVPR* 2009. 1

[5] S. Hare, A. Saffari, and P. Torr. Struck: Structured output tracking with kernels. In *ICCV*, 2011. 6

[6] G. Hua, M. Brown, and S. Winder. Discriminant embedding for local image descriptors. In *ICCV*, 2007. 5

[7] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-learning-detection. *IEEE TPAMI*, 34(7):1409–1422, 2012. 6, 7

[8] S. Leutenegger, M. Chli, and R. Y. Siegwart. Brisk: Binary robust invariant scalable keypoints. In *ICCV*, 2011. 1

[9] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60:91–110, 2004. 1

[10] G. H. M. Brown and S. Winder. Discriminative learning of local image descriptors. *IEEE TPAMI*, 33(1):43–57, 2010. 1, 3

[11] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE TPAMI*, 27(10):1615–1630, 2005. 1, 6, 7

[12] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *IJCV*, 60:(1), 63–86, 2004. 6

[13] M. Ozuysal, M. Calonder, V. Lepetit, and P. Fua. Fast keypoint recognition using random ferns. *IEEE TPAMI*, 32(3):448–461, March 2010. 4, 6, 7

[14] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In *ICCV*, 2011. 1, 4

[15] K. Simonyan, A. Vedaldi, and A. Zisserman. Learning local feature descriptors using convex optimisation. *IEEE TPAMI*, 36(8):1573–1585, 2014. 1

[16] V. L. T. Trzcinski, M. Christoudias and P. Fua. Boosting Binary Keypoint Descriptors. In *CVPR*, 2013. 1, 2, 4, 5, 6

[17] T. Trzcinski and V. Lepetit. Efficient Discriminative Projections for Compact Binary Descriptors. In *ECCV*, 2012. 1, 3

[18] T. Tuytelaars and C. Schmid. Vector quantizing feature space with a regular lattice. In *ICCV*, 2007. 2

[19] S. A. J. Winder and M. Brown. Learning local image descriptors. In *CVPR*, 2007. 1, 2, 4

[20] S. A. J. Winder, G. Hua, and M. Brown. Picking the best daisy. In *CVPR*, 2009. 1, 2

[21] Y. Wu, J. Lim, and M.-H. Yang. Online object tracking: A benchmark. In *CVPR*, 2013. 7, 8