Accumulated Stability Voting: A Robust Descriptor from Descriptors of Multiple Scales

Tsun-Yi Yang[1,2] Yen-Yu Lin[1] Yung-Yu Chuang[2]
[1]Academia Sinica, Taiwan [2]National Taiwan University, Taiwan
{shamangary,yylin}@citi.sinica.edu.tw cyy@csie.ntu.edu.tw

Abstract

This paper proposes a novel local descriptor through accumulated stability voting (ASV). The stability of feature dimensions is measured by their differences across scales. To be more robust to noise, the stability is further quantized by thresholding. The principle of maximum entropy is utilized for determining the best thresholds for maximizing discriminant power of the resultant descriptor. Accumulating stability renders a real-valued descriptor and it can be converted into a binary descriptor by an additional thresholding process. The real-valued descriptor attains high matching accuracy while the binary descriptor makes a good compromise between storage and accuracy. Our descriptors are simple yet effective, and easy to implement. In addition, our descriptors require no training. Experiments on popular benchmarks demonstrate the effectiveness of our descriptors and their superiority to the state-of-the-art descriptors.

1. Introduction

Feature matching has gained significant attention because it has a wide variety of applications such as object detection [10, 12], scene categorizing [20], image alignment [22, 17], image matching [9, 8], co-segmentation [28, 7] and many others. The performance of feature matching heavily relies on accurate feature localization and robust feature description. This paper focuses on the latter issue on feature descriptors. A good feature descriptor is often local, distinctive and invariant. Local descriptors help combating with problems of occlusion and clutter. Distinctive descriptors allow matching with high confidence. Finally, descriptors need to be invariant to various variations that one could run into in real-world applications, including orientations, scales, perspectives, and lighting conditions.

Most descriptors only represent the local neighborhood of the detected feature at a particular scale (or the domain size as called by Dong *et al*. [11]). The scale is often determined by the feature detection algorithm. The selection of the scale could be inaccurate and lead to bad matching performance. Previous methods address this issue by finding a low-dimensional linear approximation for a set of descriptors extracted at different scales [15] or performing pooling on them [11]. Inspired by these methods, we propose the accumulated stability voting (ASV) framework for synthesizing a more robust descriptor from descriptors at multiple scales. Figure 1 illustrates the framework. Our framework starts with sampling the scale space around the detected scale. With a set of baseline descriptors extracted at the sampled scales, we find the difference of two baseline descriptors for each scale pair. These differences reveal the stability (insensitivity) of each dimension of the baseline descriptor with respect to scales. By applying thresholding, we classify each dimension as "stable" or "unstable." Using multiple thresholding, the stability levels can be refined if necessary. To maximize the discriminant power of the descriptor, the principle of maximum entropy is explored for determining the best thresholds. By accumulating the stability votes from all scale pairs, we obtain a real-valued descriptor (called ASV-SIFT if SIFT is used as the baseline descriptor). The descriptor can be converted into a binary descriptor by performing the second-stage thresholding.

The main contribution of the paper is the overall design of the ASV framework. Along this way, we have (1) used differences of descriptors for measuring stability of feature dimensions across scales, (2) utilized the principle of maximum entropy for effective quantization of stability, and (3) demonstrated the effectiveness of our descriptors by thorough experiments and comparisons.

2. Related Work

Scale Invariant Feature Transformation (SIFT) [23] provides procedures for both feature detection and description with good performance. It locates features and estimates their scales by finding extrema of a Difference-of-Gaussian pyramid [24] in the scale space. A carefully designed procedure is used to collect a histogram of gradients within

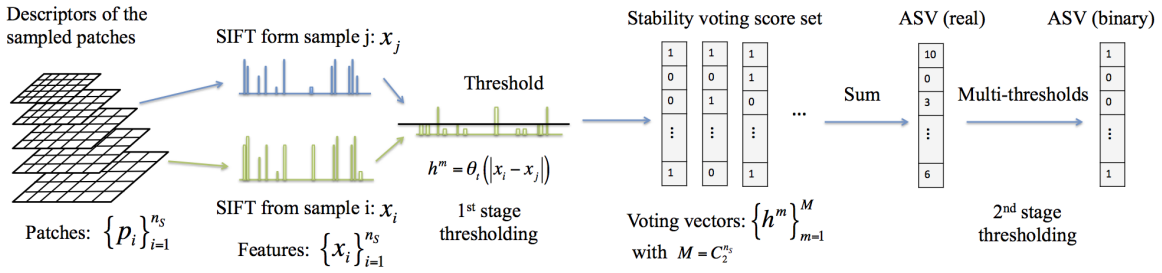

Figure 1. *The Accumulated Stability Voting (ASV) framework*. For each detected feature point, we sample patches $\{p_i\}_{i=1}^{n_s}$ at n_s different scales and extract their SIFT descriptors, $\{x_i\}_{i=1}^{n_s}$. The stability voting vectors, $\{h^m\}_{m=1}^{M}$ where $M = C_2^{n_s}$, are determined by thresholding the absolute difference between any pair of scale samples, x_i and x_j. By summing up all the stability votes, a real-valued descriptor ASV (real) is obtained. The higher the score, the more stable the corresponding dimension. A binary variant, ASV (binary), is obtained by applying multi-thresholding to the ASV (real) descriptor.

the local area around the feature. Several feature detection methods have been proposed thereafter, such as Harris-affine [25] and Hessian-affine [27] detectors. The affine approximate process [31, 25] has been proposed to refine the elliptical region detected by detectors. In addition to detectors, many real-valued descriptors have also been proposed, such as LIOP [36], DAISY [33], SURF [4], and HOG [10].

Several binary descriptors recently emerge because they are efficient in both extraction (by performing binary tests) and matching (by calculating Hamming distances). Notable binary descriptors include BRIEF [6], BRISK [21], FREAK [1], and ORB [29]. They are usually based on a set of binary tests which compare the intensity values at two predefined locations. Depending on the result of each comparison, the corresponding bit is assigned 1 or 0. The Hamming distance is usually used as the distance metric between two descriptors and its calculation is much faster than the Euclidean distance usually used for real-valued descriptors. In addition, the binary descriptors usually have compact sizes with less storage costs. Both matching efficiency and low storage requirement make binary descriptors attractive for large-scale image retrieval applications. Unfortunately, these advantages often come with the shortcoming of being less effective than real-valued descriptors.

Recently, researchers noted that one could extract the descriptor information in the real-valued domain, such as pooling of gradients or intensity values, and then encode the collected information into binary vectors. Simonyan *et al*. proposed to use a pooling region selection scheme along with dimension reduction using low-rank approximation to extract the local information, and then encode the information into a binary descriptor [32]. Trzcinski *et al*. proposed BinBoost which adopts an AdaBoost-like method for training binary descriptors using positive and negative patch pairs in the gradient domain [34]. This type of methods often utilize learning algorithms and their effectiveness usually highly depends on the training dataset. For example, BinBoost outperforms several methods in the patch dataset [5], but its performance is not as good in the other image matching dataset [18]. The same observation has been made by Balntas *et al*. [3] and our experiments. Recently, CNN-based descriptors and image matching have been proposed [13, 39]. Although effective, they require a large training set and the dimensions of the real-valued descriptors are usually very large, for example, around 4,000 for a CNN-based descriptor [13, 14].

As mentioned above, SIFT assigns each feature a detected scale and extracts the descriptor for that particular scale. Recently, Hassner *et al*. observed that selecting a single scale may lead to poor performance and suggested aggregating descriptors at multiple scale samples for improving performance [15]. They used a low-dimensional linear space to approximate a set of feature descriptors extracted at multiple scales. This descriptor is called scale-less SIFT (SLS). The similar idea has been adopted for sampling the affine space. Wang *et al*. used linear combinations to approximate affinely warped patches [37]. Balntas *et al*. adopted the similar affine sampling idea on the binary tests [3], and used a greedy method for selecting the discriminative dimensions. Similar to SLS, Domain-Size Pooling (DSP-SIFT) [11] also samples the scale space but performs pooling on the descriptors at the sample scales. Our method also explores multiple scale samples for boosting the performance of descriptors. We mostly use SIFT descriptors at multiple scales as inputs for illustrating the method. However, the proposed framework can be applied to any descriptor that can be extracted at multiple scales.

3. Accumulated Stability Voting

The proposed feature transform, Accumulate Stability Voting (ASV), accepts a multi-scale feature representation and converts it into a more robust feature descriptor.

3.1. Accumulated stability voting

Inspired by previous descriptors leveraging multi-scale SIFT representations such as SLS [15] and DSP-SIFT [11], given a SIFT feature with the detected scale σ, our method starts with obtaining a set of SIFT descriptors at several scales. The method samples n_s scales within the scale neighborhood ($\lambda_s\sigma$, $\lambda_l\sigma$) around the detected scale, where λ_s is the scaling ratio of the smallest scale and λ_l is the scaling ratio of the largest scale. At each sampled scale, a SIFT descriptor is extracted.

The philosophy of our descriptor design is to use relative information as it is regarded as more robust and invariant to variations. For a similar reason, SIFT collects statistics of gradients, the relative difference of intensity in the spatial domain. It motivates us to use the difference between SIFT descriptors for the descriptor design.

We take the absolute value of the difference between two different SIFT descriptor samples, $\mathbf{v}^{i,j} = |\mathbf{x}_i - \mathbf{x}_j|$, where $\mathbf{x}_i$ and $\mathbf{x}_j$ are SIFT descriptors extracted at two different scales i and j ($1 \leq i, j \leq n_s$) and the operator $|\cdot|$ takes the absolute value for each element of the input vector, *i.e.*, $\mathbf{v}_k^{i,j} = |\mathbf{x}_i[k] - \mathbf{x}_j[k]|$ where $k \in [1..128]$. The absolute difference reflects how stable a SIFT bin is across different scales. If the value $\mathbf{v}_a^{i,j}$ is smaller than $\mathbf{v}_b^{i,j}$, then bin a is considered more stable than bin b since bin a has more similar behavior across these two scales i and j. Thus, the vector $\mathbf{v}^{i,j}$ reflects the *stability* of each SIFT bin between scale i and scale j. There are $M = C_2^{n_s}$ possible scale pairs for n_s sampled scales. Thus, we re-index the stability vector $\mathbf{v}^{i,j}$ as $\mathbf{v}^m$ where $m \in [1..M]$. To determine how stable each SIFT bin is, one could perform pooling on these stability vectors, *i.e.*, $\mathbf{s} = \sum_{m=1}^{M} \mathbf{v}^m$. We call the vector $\mathbf{s}$ the *accumulated stability*. Each element of $\mathbf{s}$ reflects how stable the corresponding SIFT bin is across different scales. Our first attempt is to use the accumulated stability as the feature descriptor. We call it AS-SIFT where AS stands for accumulated stability. In Section 4, we show that this descriptor has slightly better performance than DSP-SIFT.

Note that our method follows the strategy of exploring and manipulating in the descriptor's scale space, laid down by previous methods, SLS, ASR, BOLD, and DSP-SIFT. These methods extract multi-scale/rotation descriptors and construct a robust descriptor by summing or forming the sub-space of them. Our method takes gradients of them. Consider an arbitrary base descriptor applied to an interest point at two different scales. Although the regions for feature extraction are not the same, the difference between the two yielded feature vectors can be interpreted as the gradient of the descriptors on that point along the scale space. Our descriptor operates on such gradients computed at multiple scales. Thus, a bin of our descriptor represents the accumulated gradient of a specific statistic, characterized by the base descriptor in that bin, along the scale space.

We take the philosophy of using relative information a step further. Instead of recording the stability value for each bin, we only classify bins as "relatively stable" or "relatively unstable." In other words, we quantize the stability value into binary. For that purpose, we perform thresholding on each stability vector and produce a binary vector $\mathbf{h}^m = \theta_t(\mathbf{v}^m | t_m)$ where θ_t is the thresholding function and t_m is the chosen threshold. Thus,

$$\mathbf{h}_d^m = \begin{cases} 1 & \mathbf{v}_d^m < t_m, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

If $\mathbf{h}_d^m = 1$, the SIFT bin d is regarded as stable for the scale pair m; otherwise it is not stable. With proper thresholding, the quantized stability could be more robust to noise. The binary stability values can be regarded as stability votes. Instead of accumulating stability values, we accumulate stability votes (or the quantized stability) for all scale pairs. This way, we obtain a real-valued descriptor $\mathbf{C}$:

$$\mathbf{C} = \sum_{m=1}^{M} \mathbf{h}^m = \sum_{m=1}^{M} \begin{bmatrix} \mathbf{h}_1^m \\ \mathbf{h}_2^m \\ \vdots \\ \mathbf{h}_{128}^m \end{bmatrix}. \tag{2}$$

We call the descriptor ASV-SIFT where ASV stands for *accumulated stability voting*.

3.2. Local threshold determination

The remaining issue is how to determine the proper threshold value t_m for each scale pair m. One possibility would be to learn a global threshold value from the statistics of a set of training examples. Instead of a global threshold, we determine a local threshold for each pair of scales using the principle of maximum entropy for maximizing information (the discriminant power in our case). Entropy is a metric for measuring uncertainty or the amount of information. In the field of image thresholding, *the principle of maximum entropy* or *maximum entropy modeling* has been used for determining proper thresholds [19, 38, 30].

In our case, we want to find a threshold to maximize the entropy of the resultant binary voting vector. From the definition of entropy, the maximal information occurs when the number of 1s equals the number of 0s in the resultant vector. Thus, the optimal threshold $t_m^\star$ would be the median of all elements in $\mathbf{v}^m$, *i.e.*, $t_m^\star = median(\mathbf{v}_1^m, \mathbf{v}_2^m, \cdots, \mathbf{v}_{128}^m)$. There are more complex methods for determining thresholds. However, our experiments found that the median is simple yet effective. With this rule, the threshold can be determined locally for each scale pair.

3.3. Multiple thresholding

Using only binary quantization levels could be too restrictive and reduce the discriminative power of the descriptor. We can increase the number of quantization levels by

determining multiple thresholds. Although there are quite a few multiple-threshold selection schemes [19], we found that the same maximum entropy principle can still be applied. We denote the number of thresholds as n_t^1 for the first-stage thresholding. Multiple thresholding would partition the range of stability values into (n_t^1+1) quantization levels. By applying the principle of maximum entropy again, the resultant vector must contain the same number of elements for all groups and the thresholds can be determined thereby. For n_t^1 thresholds, the maximal number of votes (the quantized stability value) from each pair of scales is n_t^1. Thus, the maximal number of votes a bin can receive is $C_2^{n_s} \times n_t^1$ and would require $\lceil \log_2 \left(C_2^{n_s} \times n_t^1\right) \rceil$ bits to store. Taking $n_t^1 = 7$ and $n_s = 10$ as an example, there are 8 quantization levels and the maximal number of votes from each scale pair is 7. Thus, the maximal votes a bin can receive is $45 \times 7 = 315$ and it would take 9 bits for each bin, making the size of the ASV-SIFT descriptor 128×9 bits.

3.4. Feature interpolation

In principle, the more scale samples we obtain, the more information we have. However, in our case, taking more scale samples means extracting more SIFT descriptors. The overall extraction time depends linearly on the number of scales. For obtaining more samples without incurring too much computation overhead, we interpolate SIFT feature descriptors at the neighboring scales into the additional feature descriptor at the intermediate scale, $\mathbf{x}'_i = \frac{\mathbf{x}_i + \mathbf{x}_{i+1}}{2}$, where $i \in [1..n_s - 1]$. This way, the number of sample features we can utilize nearly double, $n'_s = n_s + (n_s - 1)$. Experiments show that the feature interpolation strategy consistently improves the performance of the proposed framework, especially when the sample number is very small, for example, $n_s = 4$.

3.5. Second stage thresholding

The ASV-SIFT descriptor we have discussed so far is a real-valued descriptor. In some applications, binary descriptors are preferred. For converting ASV-SIFT into a binary descriptor, the second-stage thresholding can be added for quantizing the real-valued descriptor. Assume that we have obtained a real-valued ASV-SIFT descriptor with the parameters n_s and n_t^1, as discussed above, the maximal value for each dimension is $C_2^{n_s} \times n_t^1$. For applying the second-stage thresholding, we first decide how many thresholds are used and denote it as n_t^2. To divide the range $\left[0, C_2^{n_s} \times n_t^1\right]$ into $n_t^2 + 1$ intervals equally, the k-th threshold would be $\left\lfloor \frac{C_2^{n_s} \times n_t^1 \times k}{n_t^2 + 1} \right\rfloor$.

4. Experimental Results

In this section, we first describe the datasets and detector setting. Next, we list the methods for comparisons and then examine the parameter setting. Finally, we compare performance of descriptors and analyze the results.

4.1. Datasets and detector setting

We used three datasets for comparing feature descriptors: the Oxford dataset [18], the Fischer dataset [13], and the local symmetry dataset (SYM dataset [16]). The Oxford dataset is a popular matching dataset and it contains 40 image pairs under different variations including blurring, compression, viewpoints, lighting and others. The Fischer dataset was introduced to address the deficiencies of the Oxford dataset. It provides 400 image pairs with more extreme variations such as zooming, rotation and perspective and non-linear transformations. The SYM dataset contains 46 image pairs of architectural scenes. It is a challenging matching dataset as the images exhibits dramatic variations in lighting, age and even styles.

For each dataset, we choose the detector with the state-of-the-art performance to better reflect the scenarios of the descriptor's use in real-world applications. For the Oxford dataset and the Fischer dataset, we use the Difference-of-Gaussian detector with affine approximation (DoGAff)[1], implemented in VLFeat. We sort the detected features by the peakScores of VLFeat's *covdet* function [35] and choose around 5,000 feature points per image. For the SYM dataset, we adopt SYM-I and SYM-G as the detectors [16]. These three datasets provide ground-truth transformations so that quantitative analysis is possible. We follow the same evaluation procedure proposed by Mikolajczyk and Schmids [26]. A feature match is evaluated by calculating the intersection-over-union (IoU) of the areas of the ground-truth match and the detected match. It is considered correct if overlapping $>50\%$. By varying the threshold on the distance between descriptors which are considered a match, one can obtain the precision-recall curve. The area under the curve is called average precision (AP). The average of APs, mAP, is used for comparing performance of detectors.

4.2. Descriptors for comparisons

We compare our descriptors with the following descriptors. (1) **SIFT** [23]. We use VLFeat toolbox for extracting SIFT descriptors. (2) **Root-SIFT** [2]. It reformulates the SIFT distance by using Hellinger kernel. (3) **ASR** [37]. Similar to scale-less SIFT (SLS) [15], it represents a set of the affine warped patches by a subspace representation. (4) **DSP-SIFT** [11]. The Domain-Size Pooling (DSP) operation is performed on SIFT descriptors at different scales. We used the source code provided by authors for obtaining the descriptors for features while VLFeat's covariant detector is used for detecting features. (5) **RAW-PATCH** [35]. It is the raw patch descriptor extracted by VLFeat. The size of

[1]Note that Dong and Soatto used MSER as the detector in their experiments [11]. We use DoGAff because of its much better performance.

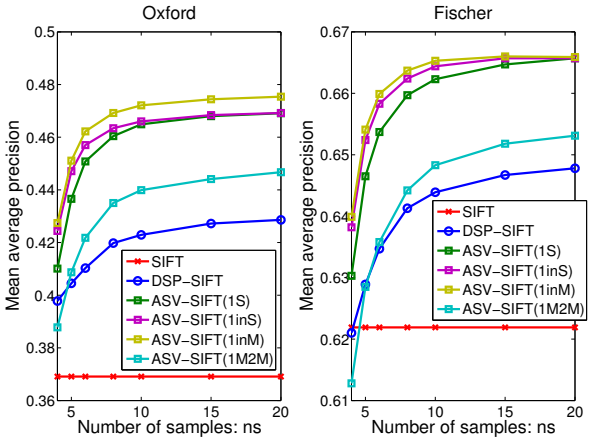

Figure 2. *Experiments with the number of sampled scale* n_s. The results for the Oxford dataset are on the left, and the results for the Fischer dataset are on the right. For all methods other than SIFT, the performance improves with more scale samples. However, the performance gain nearly saturates after 10 samples. Note that 1inS and 1inM mean that feature interpolation is used. The interpolation strategy helps especially when n_s is small.

the patch is fixed at 41×41. (6) **LIOP** [36]. The intensity order based descriptor extracted by VLFeat. (7) **SYMD** [16]. The descriptor is designed specifically for capturing locally symmetry. The concatenation of our descriptor and SYMD descriptor is denoted as ASV-SIFT (1S*):SYMD. (8) **BOLD** [3]. One of the state-of-the-art binary descriptors. It uses a local mask to preserve the stable dimensions. (9) **BinBoost** [34]. A binary descriptor learned by a boosting method.

Next, we describe the notations for different variants of our descriptor. (1) **AS-SIFT**. A variant of our descriptor by accumulating the stability vectors $\mathbf{v}^m$ instead of voting vectors $\mathbf{h}^m$. The parameter setting is the same as **ASV-SIFT (1S*)**. (2) **ASV-SIFT (1S*)**. A real-valued descriptor after our first stage thresholding and accumulation. 1S means the first stage single thresholding. The asterisk denotes the specific parameter setting used in Table 1. (3) **ASV-SIFT (1M2M*)**. Our binary descriptor after the two-stage process. 1M2M means that multi-thresholding is used in both the first and the second stages. The asterisk denotes the specific parameter setting used in Table 2. (4) **ASV-PATCH**, **ASV-LIOP**. Real-valued descriptors by applying our ASV framework to RAW-PATCH and LIOP descriptors.

4.3. Parameter setting

Similar to DSP-SIFT, our method has three parameters for determining the sampled scales: λ_s for scaling ratio of the smallest scale, λ_l for scaling ratio of the largest scale, and n_s for the number of sampled scales in between. We performed experiments to find the best parameters of both our ASV framework and DSP-SIFT.

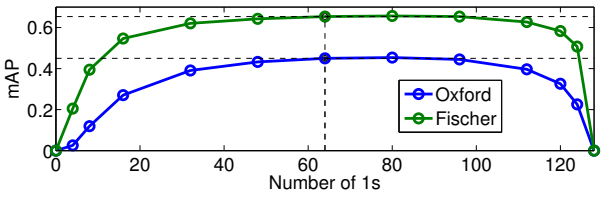

Figure 3. *Validation of the maximum-entropy-principle-based threshold selection scheme.* The number of 1s of each voting vector $\mathbf{h}^m$ is determined by the chosen threshold. The black dash line indicates where the best mAP values occur. It shows our median thresholding scheme maximizes the information of descriptors, thus leading to the best performance among different threshold choices.

Figure 2 shows the mAP values for both Oxford and Fischer datasets with different numbers of sampled scales n_s. Not surprisingly, the performance of all multi-scale methods improves when n_s increases. It is because more scale samples usually carry on more information. However, the performance gain becomes marginal when there are too many samples. Note that our real-valued descriptor ASV-SIFT (1S) outperforms DSP-SIFT with 20 sampled scales by only using 6 sampled scales. Our binary descriptor ASV-SIFT (1M2M) also outperforms DSP-SIFT with 20 samples when n_s reaches 10. This figure also shows the effects of feature interpolation. ASV-SIFT (1inS) in Figure 2 represents ASV-SIFT (1S) with feature interpolation, similarly for ASV-SIFT (1inM). The figure shows that the interpolation strategy does help especially when n_s is small.

Note that, although our ASV framework requires thresholding, the threshold values are dynamically and locally determined without external training or applying hashing. Thus, there is no extra parameter for thresholding. To validate our threshold selection scheme based on the maximum entropy principle, we examine the matching performance with different numbers of 1s and 0s in the voting vector after thresholding. Figure 3 shows mAP values with different numbers of 1s after binary thresholding. The black dot line indicates where the best mAP values locate. For both Oxford and Fischer datasets, the best mAP occurs when the numbers of 1s and 0s are equal to each other.

In the following experiments, we empirically choose $\lambda_s = \frac{1}{6}$, $\lambda_l = 3$, and $n_s = 10$ for DSP-SIFT[2], ASV-SIFT (1S*), and ASV (1M2M*). Note that we do not use feature interpolation in the following experiments as n_s is sufficiently large. For ASV-SIFT, when multiple thresholding is used in either the first or the second stage, we also need to determine the numbers of thresholds, n_t^1 and n_t^2 . We will discuss these parameters in the next section. For other descriptors, we use the parameters suggested by the original paper or the implementation.

[2]Note that, because a different detector is used, the empirically chosen parameters for DSP-SIFT are different from the original paper [11].

Method	Bits	mAP	
		Oxford	Fischer
real-valued descriptor			
SIFT	128*8	0.3691	0.6219
DSP-SIFT	128*8	0.4229	0.6439
Root-SIFT	128*8	0.4218	0.6504
ASR	300*8	0.2470	0.5252
AS-SIFT	128*8	0.4298	0.6447
RAW-PATCH	1681*8	0.1344	0.4063
LIOP	144*8	0.1543	0.5009
1st stage single thresholding ($1S$)			
ASV-SIFT (n_s=10)*	128*6	**0.4649**	**0.6623**
ASV-SIFT (n_s=10, pP)	128*6	0.4479	0.6565
ASV-PATCH (n_s=10)	1681*6	0.3560	0.6118
ASV-LIOP (n_s=10)	144*6	0.4124	0.6457
1st stage multiple thresholding ($1M$)			
ASV-SIFT (n_s=10, n_t^1=3)*	128*8	0.4731	0.6630
ASV-SIFT (n_s=10, n_t^1=7)	128*9	0.4739	**0.6631**
ASV-SIFT (n_s=10, n_t^1=15)	128*10	**0.4740**	0.6627

Table 1. The comparisons of real-valued descriptors.

Method	Bits	mAP	
		Oxford	Fischer
binary descriptor			
BOLD	512*1	0.2937	0.5532
BinBoost	256*1	0.1815	0.3553
$1S^\star$ with 2nd stage single thresholding ($1S2S$)			
ASV-SIFT (n_s=10)	128*1	0.3517	0.5855
$1M^\star$ with 2nd stage single thresholding ($1M2S$)			
ASV-SIFT (n_s=10)	128*1	0.3587	0.5945
$1M^\star$ with 2nd stage multiple thresholds ($1M2M$)			
ASV-SIFT (n_s=10, n_t^2=2)	256*1	0.4234	0.6369
ASV-SIFT (n_s=10, n_t^2=3)*	384*1	**0.4399**	**0.6483**

Table 2. The comparisons of binary descriptors.

Method	mAP	
	SYM-I	SYM-G
SIFT	0.3522	0.4077
SYMD	0.2608	0.3276
DSP-SIFT	0.3679	0.4208
ASV-SIFT(1S*)	**0.3734**	**0.4345**
SIFT:SYMD	0.3692	0.4553
DSP-SIFT:SYMD	0.3885	0.4627
ASV-SIFT(1S*):SYMD	**0.4039**	**0.4738**

Table 3. mAP for the SYM dataset

4.4. Quantitative evaluation

Table 1 and Table 2 respectively report the matching performance of real-valued descriptors and binary descriptors on both Oxford and Fischer datasets. In addition to mAP, the two tables also report the storage requirements for all descriptors. For single-stage single thresholding ASV-SIFT (1S), the size of each dimension depends on the number of sampled scales. For our implementation in which $n_s = 10$, the number of scale pairs is $C_2^{n_s} = 45$ and the range of each dimension in ASV-SIFT (1S*) is $[0..45]$. Thus, 6 bits are required for storing each dimension. For single-stage multiple thresholding ASV-SIFT (1M), in addition to n_s, the size of each dimension also depends on how many quantization levels we choose. For example, if 8 levels are used and $n_s = 10$, the maximal value of a dimension is $45 \times (8-1) = 315$ and 9 bits are required for each dimension. As shown in Table 1, although more quantization levels could slightly improve mAP, for making the descriptors more compact, we choose 4 levels (with 3 thresholds, $n_t^1 = 3$) as the best parameter for ASV-SIFT (1M*). Finally, for the second-stage multiple thresholding, we choose to use 4 quantization levels (with 3 thresholds, $n_t^2 = 3$) as the best parameter for ASV-SIFT (1M2M*) because it improves mAP significantly with only the modest size increase. Note that each dimension of the real-valued descriptors is assumed to be stored in a byte as higher precision only leads to minor improvement.

Similar to [11], in our experiments, DSP-SIFT outperforms SIFT by a margin. Root-SIFT has similar performance with DSP-SIFT. ASR performs poorly. It is probably because the regions detected by DoGAff are relatively small and the affine samples might not be useful for the ASR method. After the first-stage thresholding, our ASV-SIFT (1S*) consistently outperforms other real-valued descriptors in both Oxford and Fischer datasets. The asterisk in the table indicates the specific parameter setting we mentioned earlier. In Table 1, we also show mAP when using another threshold selection scheme [19] (indicated by ASV-SIFT ($n_s = 10$, pP) in Table 1). It shows that the more complex threshold selection schemes may not lead to better performance. The necessity of the first stage thresholding is confirmed by comparing AS-SIFT and ASV-SIFT in Table 1 and also shown in the third column of the head-to-head comparisons in Figure 6. Our ASV framework can also be applied to descriptors other than SIFT. We applied it for converting RAW-PATCH into ASV-PATCH, and LIOP into ASV-LIOP. Table 1 shows the performance is significantly improved with the ASV framework. Table 2 compares two state-of-the-art binary descriptors, BOLD and BinBoost, with our binary descriptor using the second-stage thresholding. By using multiple thresholding in both the first and the second stages, our ASV-SIFT (1M2M*) defeats the other binary descriptors by a great margin. It even outperforms many real-valued descriptors such as SIFT, ASR, and DSP-SIFT.

The SYM dataset has its own characteristics and different detectors are used for the state-of-the-art performance. Table 3 summarizes the matching performance of several

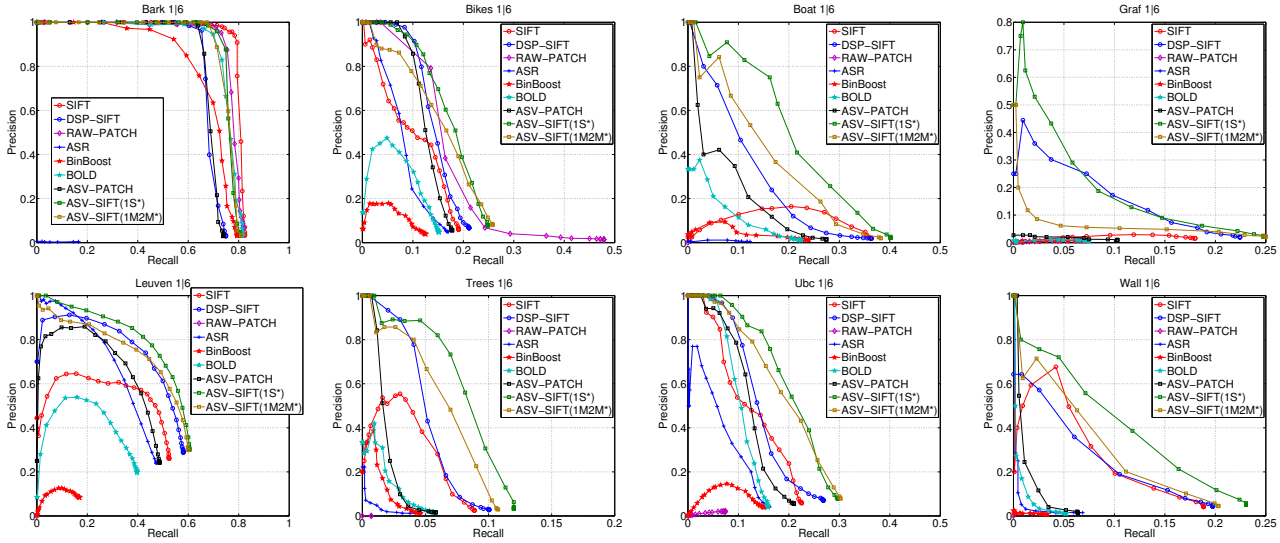

Figure 4. *PR curves of challenging pairs in the Oxford dataset.* The most challenging pair (magnitude 6 in Figure 5) of each class is presented in the figure.

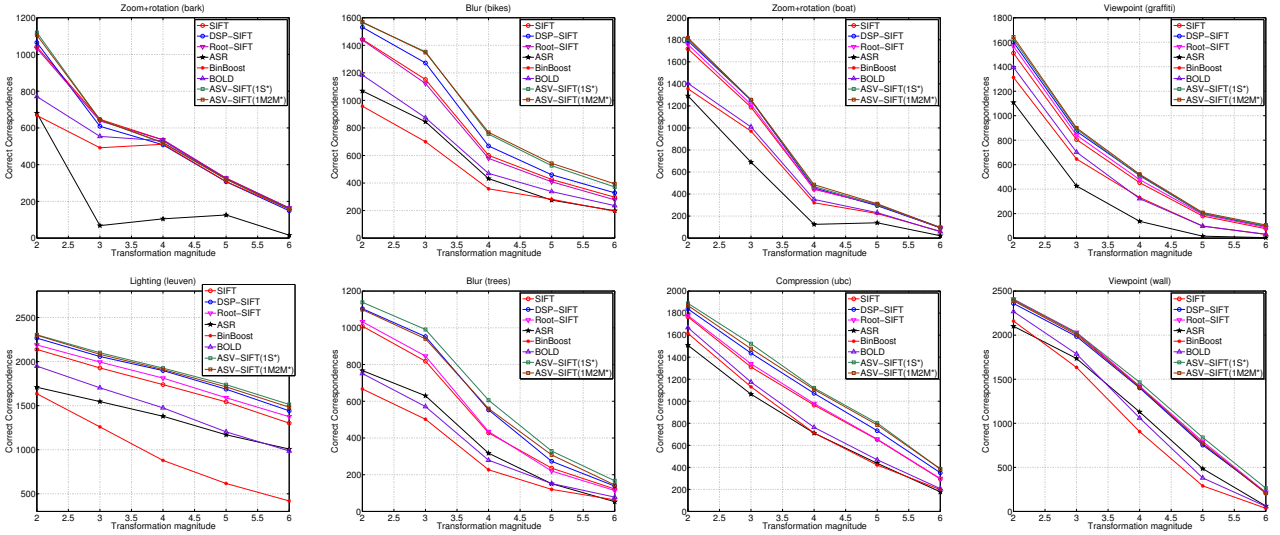

Figure 5. *The number of correct correspondences using different descriptors under different transformation magnitudes in the Oxford dataset*. The weakest magnitude is 2, and 6 is the strongest one. The performance generally degrades when the magnitude gets stronger.

descriptors with two different detectors, SYM-I and SYM-G. In either case, ASV-SIFT outperforms SIFT, DSP-SIFT and SYMD, a descriptor specially designed for the scenario of the dataset. As indicated by Hauagge and Snavely [16], the performance of a descriptor can be boosted by combining with the SYMD descriptor as they explore complementary information. After a weighted concatenation with the SYMD descriptor, ASV-SIFT (1S*):SYMD has the best performance for this dataset.

For more careful examination, Figure 4 displays the precision-recall (PR) curves of several most challenging image pairs in the Oxford dataset. Our ASV-SIFT (1S*) and ASV-SIFT (1M2M*) descriptors perform the best in most cases. Particularly, our method is robust to extreme transformations such as the ones in Boat (1|6) (zoom+rotation) and Graf (1|6) (viewpoint). In Figure 5, the numbers of correct correspondences by using different descriptors are compared under different transformation magnitudes. Our descriptors are particularly good at finding the correct matching pairs for blur and compression transformations. For other transformations such as rotation, viewpoint, and zoom, although without clear advantages, our descriptors keep pace with top performers. ASR [37] and BOLD [3] cannot perform well probably because the detected region

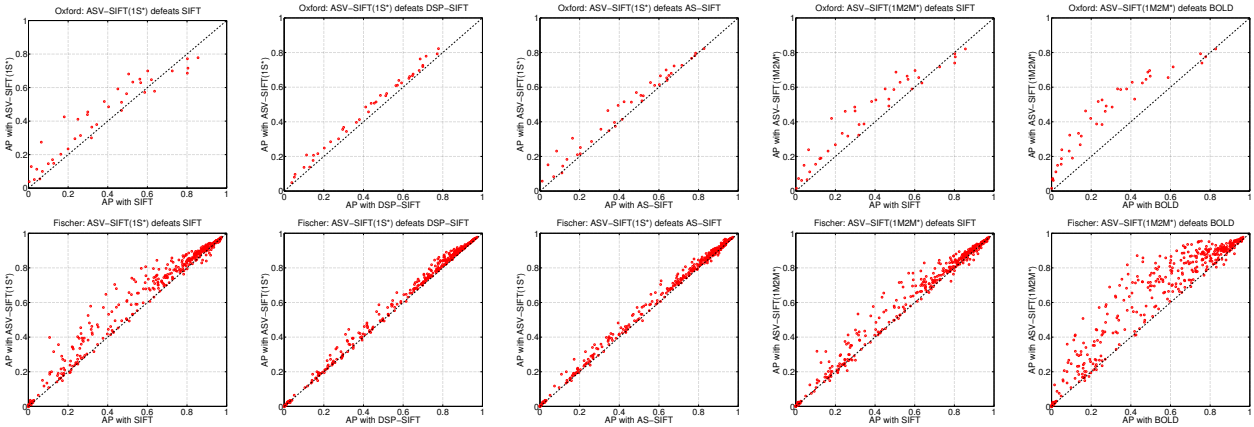

Figure 6. *Head-to-head comparisons*. The first row is for the Oxford dataset while the second row is for the Fischer dataset. The first three columns compare our real-valued descriptor ASV-SIFT (1S⋆) with SIFT, DSP-SIFT and AS-SIFT respectively. The last two compare our binary descriptor ASV-SIFT (1M2M⋆) with SIFT and BOLD.

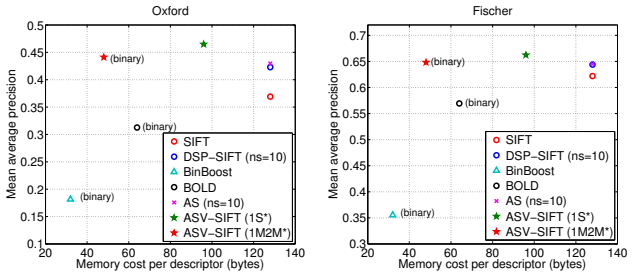

Figure 7. *Comparisons of descriptors in terms of both storage requirement and matching performance*. Our real-valued descriptor ASV-SIFT (1S⋆) provides the best matching performance among all descriptors. Our binary descriptor ASV-SIFT (1M2M⋆) gives comparable performance to state-of-the-art real-valued descriptors with only 46 bytes per feature.

is small and under the zoom or viewpoint transformation, exploring rotation samples does not help much. The bad performance of BinBoost [34] could be attributed to the different training dataset.

Figure 6 shows the head-to-head comparisons between two selected descriptors. Each point in the plot displays the average precision (AP) of an image pair using the two selected descriptors. The first two columns shows that our real-valued descriptor ASV-SIFT (1S⋆) outperforms SIFT and DSP-SIFT on most image pairs of both Oxford and Fischer datasets. The third column compares ASV-SIFT (1S⋆) and AS-SIFT. It shows that the first-stage thresholding consistently improves the matching performance. The last two columns show that our binary descriptor ASV-SIFT (1M2M⋆) is superior to SIFT (even though SIFT consumes much more space) and the state-of-the-art binary descriptor BOLD by a large margin.

4.5. Storage and time requirements

Figure 7 plots descriptors as a point in the space of storage requirement and matching performance. Our real-valued descriptor ASV-SIFT (1S⋆) provides the best matching performance among all descriptors. With only 46 bytes per feature, our binary descriptor ASV-SIFT (1M2M⋆) gives comparable performance to DSP-SIFT while significantly outperforming SIFT, BOLD, and BinBoost. It shows that the two-stage thresholding strategy is quite effective on making a good compromise between storage requirement and matching performance. Similar to DSP-SIFT, the extraction time of the ASV-SIFT descriptor depends linearly to the number of sampled scales. On average, our method takes 10.2 (21.0) seconds for constructing a descriptor for an image in the benchmarks with 4 (10) sampled scales.

5. Conclusion

We have introduced a new local descriptor: accumulated stability voting. It provides both a real-valued descriptor and a binary descriptor. The real-valued descriptor outperforms the state-of-the-art descriptors in terms of mAP. The binary descriptor consumes only about one-third of storage while still maintaining a decent matching performance, making a good compromise between storage requirement and matching effectiveness. More results and the codes of this framework are available at https://github.com/shamangary/ASV.

Acknowledgement

This work was supported by Ministry of Science and Technology (MOST) under grants 103-2221-E-001-026-MY2, 104-2628-E-001-001-MY2, and 104-2628-E-002-003-MY3.

References

[1] A. Alahi, R. Ortiz, and P. Vandergheynst. FREAK: Fast Retina Keypoint. In *CVPR*, 2012.

[2] R. Arandjelovic and A. Zisserman. Three Things Everyone Should Know to Improve Object Retrieval. In *CVPR*, 2012.

[3] V. Balntas, L. Tang, and K. Mikolajczyk. BOLD - Binary Online Learned Descriptor For Efficient Image Matching. In *CVPR*, 2015.

[4] H. Bay, T. Tuytelaars, and L. V. Gool. SURF: Speeded Up Robust Features. In *ECCV*, 2006.

[5] M. Brown, G. Hua, and S.Winder. Discriminative Learning of Local Image Descriptors. *TPAMI*, 2011.

[6] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. BRIEF: Binary Robust Independent Elementary Features. In *ECCV*, 2010.

[7] K. Chang, T. Liu, and S. Lai. From Co-saliency to Co-segmentation: An Efficient and Fully Unsupervised Energy Minimization Model. In *CVPR*, 2011.

[8] H. Chen, Y. Lin, and B. Chen. Robust Feature Matching with Alternate Hough and Inverted Hough Transforms. In *CVPR*, 2013.

[9] M. Cho, J. Lee, and K. M. Lee. Deformable Object Matching via Agglomerative Correspondence Clustering. In *ICCV*, 2009.

[10] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In *CVPR*, 2005.

[11] J. Dong and S. Soatto. Domain-Size Pooling in Local Descriptors: DSP-SIFT. In *CVPR*, 2015.

[12] P. F. Felzenszwalb, R. B. Girshick, D. A. McAllester, and D. Ramanan. Object Detection with Discriminatively Trained Part-Based Models. *TPAMI*, 2010.

[13] P. Fischer, A. Dosovitskiy, and T. Brox. Descriptor Matching with Convolutional Neural Networks: A Comparison to SIFT. *arXiv*, 2014.

[14] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. Berg. MatchNet: Unifying Feature and Metric Learning for Patch-Based Matching. In *CVPR*, 2015.

[15] T. Hassner, V. Mayzels, and L. Zelnik-Manor. On SIFTs and their Scales. In *CVPR*, 2012.

[16] D. C. Hauagge and N. Snavely. Image Matching using Local Symmetry Features. In *CVPR*, 2012.

[17] K. Hsu, Y. Lin, and Y. Chuang. Robust image alignment with multiple feature descriptors and matching-guided neighborhoods. In *CVPR*, 2015.

[18] G. Hua, M. Brown, and S.Winder. Discriminant Embedding for Local Image Descriptors. In *ICCV*, 2007.

[19] J. N. Kapur, S. P. K., and W. a. K. C. A New Method for Gray-Level Picture Thresholding Using the Entropy of the Histogram. *CVGIP*, 1985.

[20] S. Lazebnik, C. Schmid, and J. Ponce. Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In *CVPR*, 2006.

[21] S. Leutenegger, M. Chli, and R. Y. Siegwart. BRISK: Binary Robust Invariant Scalable Keypoints. In *ICCV*, 2011.

[22] C. Liu, J. Yuen, and A. Torralba. SIFT Flow: Dense Correspondence across Scenes and Its Applications. *PAMI*, 2011.

[23] D. G. Lowe. Object Recognition from Local Scale-Invariant Features. In *ICCV*, 1999.

[24] D. G. Lowe. Distinctive Image Features from Scale-Invariant Key-points. *IJCV*, 2004.

[25] K. Mikolajczyk and C. Schmid. An Affine Invariant Interest Point Detector. In *ICCV*, 2002.

[26] K. Mikolajczyk and C. Schmid. A Performance Evaluation of Local Descriptors. *TPAMI*, 2005.

[27] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool. A Comparison of Affine Region Detectors. *IJCV*, 2005.

[28] J. C. Rubio, J. Serrat, A. Lopez, and N. Paragios. Unsupervised Co-segmentation Through Region Matching. In *CVPR*, 2012.

[29] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: An Efficient Alternative to SIFT or SURF. In *ICCV*, 2011.

[30] P. K. Saha and J. K. Udupa. Optimum Image Thresholding via Class Uncertainty and Region Homogeneity. *TPAMI*, 2001.

[31] F. Schaffalitzky and A. Zisserman. Multi-view Matching for Unordered Image Sets, or "How Do I Organize My Holiday Snaps?". In *ECCV*, 2002.

[32] K. Simonyan, A. Vedaldi, and A. Zisserman. Descriptor Learning Using Convex Optimisation. In *ECCV*, 2012.

[33] E. Tola, V. Lepetit, and P. Fua. DAISY: An Efficient Dense Descriptor Applied to Wide-Baseline Stereo. *TPAMI*, 2010.

[34] T. Trzcinski, M. Christoudias, P. Fua, and V. Lepetit. Boosting Binary Keypoint Descriptors. In *CVPR*, 2013.

[35] A. Vedaldi and B. Fulkerson. VLFeat - An Open and Portable Library of Computer Vision Algorithms. In *ACM MM*, 2010.

[36] Z. Wang, B. Fan, and F. Wu. Local Intensity Order Pattern for Feature Description. In *ICCV*, 2011.

[37] Z. H. Wang, B. Fan, and F. C. Wu. Affine Subspace Representation for Feature Description. In *ECCV*, 2012.

[38] A. K. C. Wong and P. K. Sahoo. A Gray-Level Threshold Selection Method Based on Maximum Entropy Principle. *TSMC*, 1989.

[39] S. Zagoruyko and N. Komodakis. Learning to compare image patches via convolutional neural networks. In *CVPR*, 2015.