

Compressed Binary Discriminative Feature for Fast UAV Image Registration

Li-Chuan Geng^{1,2(✉)}, Ze-xun Geng^{1,2,3}, and Guo-xi Wu^{1,2}

¹ School of Urban Planning and Landscaping,
Xuchang University, Xuchang, China
glch1982@163.com

² The UAVLRS Collaborative Innovation Center,
Xuchang University, Xuchang, China

³ The PLA Information Engineering University, Zhengzhou, China

Abstract. Efficiently UAV images mosaicking is of critical importance for the application of disaster management, in which fast image registration plays an important role. Towards fast and accurate image registration, the key design lies in the keypoint description, to which end SIFT and SURF are widely leveraged in the related literature. However, the expensive computation and memory costs restrict their potential in disaster management. In this paper, we proposed a novel keypoint descriptor termed CBDF (Compressed Binary Discriminative Feature). A cascade of binary strings is computed by efficiently comparing image gradients static information over a log-polar location grid pattern. Extensive evaluations on benchmark datasets and real-world UAV images show that CBDF yields a similar performance with SIFT and SURF, and it is much more efficient in terms of both computation time and memory.

Keywords: Image mosaic · Image registration · Unmanned aerial vehicles (UAV) · CBDF

1 Introduction

Unmanned Aerial Vehicles (UAV) is widely used in civilian applications. Comparing to standard airborne aerial, UAV system is more flexible, efficient, especially for small area coverage [1], which is especially suitable for time critical events where rapidly acquiring current and accurate spatial information is critically important. In such a case, a large number of UAV images should be processed, e.g. building an image mosaic, in a short time, with moderate accuracy, a near-orthophoto accuracy. However, the traditional photogrammetry and automatic aerial triangulation (AAT) cannot efficiently create such mosaic due to the high variation of UAV images.

One of the approaches is to decompose an image into local regions of interest, or so-called ‘features’ to alleviate the time complexity. To this end, the past decades have seen considerable advances in feature descriptors and matching strategy. Representative approaches include SIFT [2, 3] and its variants, which are distinctive and invariant to various image transformations. However, since SIFT is computationally expensive,

alternative methods such as SURF [4], PCA-SIFT [5] have been proposed to speed up. These methods have similar matching rates with SIFT while much faster performance.

Unfortunately, these gradient based methods still hard to meet the requirement of real-time image mosaic, especially on mobile devices with low computing power and memory capacity. As a result, algorithms with fixed-point operations and low memory load are preferred, such as Binary Robust Independent Elementary Feature (BRIEF) [7], Oriented Fast and Rotated BRIEF (ORB) [8], Binary Robust Invariant Scalable Keypoints (BRISK) [9], and Fast Retina Keypoint (FREAK) [10]. Such hand-crafted and heavily-engineered features are difficult to generalize to new domains. Other works [14, 20] learn short binary codes by minimizing the distances between positive training feature descriptors, while maximizing the negative pairs. Binarization is usually performed by multiplying the descriptors by a projection matrix, subtracting a threshold vector, and retaining only the sign of the result. Although low memory load, these new descriptors tend to perform worse than the floating point descriptors.

Aiming to bridge this performance gap between floating descriptors and binary descriptors without increasing the computation cost, a Compressed Binary Discriminative Feature (CBDF) is proposed for fast image registration. First, a Gaussian smoothing is applied to the image patch around the keypoint, and then local image gradients are computed. Second, the image patch is split up into smaller sub-regions, similar to GLOH [6], and a vector of the gradient statistic information is calculated for each sub-region, our binary descriptor is computed by a comparison of these vectors. Since low bit-rate feature descriptor means fast to match and low memory footprint, our goal is to produce low bit-rate descriptors which maintain the highest possible fidelity. We optimize our descriptor by a supervised learning method to find the dimensions in the descriptor which are informative to the descriptor. The advantage of our feature descriptor is that the gradient information is contained in our binary keypoint descriptor, which makes our binary descriptor much more discriminative than the simple pixel intensity comparison descriptors, in the meantime, a learning process is performed to realize dimension reduction, and this process makes our descriptor more compact.

The rest of this paper is organized as follows: The related work is presented in Sect. 2. The method to construct our CBDF descriptor is described in details in Sect. 3. In Sect. 4, we compare the performances of our CBDF descriptor with the state-of-the-art methods. Finally, we conclude our work in Sect. 5.

2 Related Work

Due to the fast development of UAV system, low altitude remote sensing is becoming more and more attractive for commercial and military applications. Such technique, if applicable, can be widely used in earthquake relief work, forest fire surveillance and flood disaster.

In such a scenario, one important step is to mosaic the UAV image in real-time. Here, one key technique is image registration. Different from the state-of-the-art image

registration algorithms like SIFT matching [11], where the registration accuracy is of fundamental importance, under the UAV circumstance the efficiency is more important. To speed up, several approaches are proposed in the literature like PCA-SIFT [5], SURF [4], CHOG [21], DAISY [23], etc. For instance, PCA-SIFT reduces the description vector from 128 to 36 dimension using principal component analysis. The matching time is reduced, but the time to build the descriptor is increased leading to a small gain in speed and a loss of distinctiveness. The SURF descriptor sums responses of Haar wavelets, which is fast by using integral image. SURF addresses the issue of speed. However, since the descriptor is a 64-vector of floating point, its representation still requires 256 bytes, which becomes crucial when millions of descriptors must be stored. Chandrasekhar et al. [21] applies tree-coding method for lossy compression of probability distributions to SIFT-like descriptors to obtain compressed histogram of gradients (CHOG). Brown et al. [22] use a training method to optimize the filtering and normalization steps that produce a SIFT-like vector. However, the dimensionality of the feature vector is still too high for large-scale applications, such as image retrieval or 3D reconstruction.

Much research has been done recently focusing on designing binary descriptor to reduce both the matching time and storage cost [12, 13]. For example, Calonder et al. [7] show that it is possible to shortcut the dimensionality reduction step by directly building a short binary descriptor in which each bits are independent, called BRIEF. The descriptor vector is obtained by comparing the intensity of 512 pairs of pixels or even 256 pairs after applying a Gaussian smoothing. The smoothing step is to reduce the noise sensitivity. The positions of the pixels are pre-selected randomly according to a Gaussian distribution or Uniform distribution around the patch center. However, this descriptor is not invariant to scale and rotation changes. Rublee et al. [8] propose the Oriented Fast and Rotated BRIEF (ORB) descriptor. This binary descriptor is invariant to rotation, which is robust to noise but not invariant to scale change while relying on a greedy optimization. Leutenegger et al. [9] propose a binary descriptor called BRISK, which is invariant to scale and rotation. Its key design lies in the application of a novel scale-space. To build the descriptor bit-stream, a limited number of points in a polar sampling pattern are used. Each point contributes many pairs. The pairs are divided into short-distance and long-distance subsets. The long-distance subset is used to calculate the direction of the keypoint, while the short-distance subset is used to build the binary descriptor. Alahi et al. [10] propose a keypoint descriptor termed Fast Retina Keypoint (FREAK), which is inspired by the human retina topology. A cascade of binary strings are computed by comparing image intensities efficiently over a retinal sampling pattern. Strecha et al. [20] map the SIFT descriptor vectors into the Hamming space by a LDA method. Although these binary feature descriptors are fast to compute and match, they tend to be less robust than their floating point equivalents.

3 The Method

Most of the local image descriptors are extracted based on the keypoints returned by interest point detectors, such as Harris [17], DoG [2], MSER [18], Hessian-Affine [19] or FAST [15]. Our proposed CBDF descriptor can be combined with any of these local

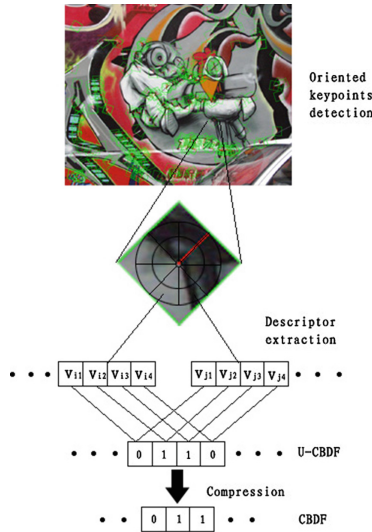


Fig. 1. Overview of the computation of CBDF descriptor

feature detectors. As finding a rotation-invariant and efficient detector is important to image registration, especially in the aerial images. So here we take the FAST as our keypoint detector. Unfortunately, FAST does not consider the orientation of the keypoint, so we proposed an enhanced-version of FAST, called orientation FAST. After oriented keypoints extraction, we build our CBDF descriptor by dividing the image patch surrounding the keypoints into subregions, and then we static the gradient information for each subregion, a four dimension vector is acquired. CBDF descriptor is constructed by comparing and thresholding these vectors. At last a dimension reduction scheme is applied on the descriptor to make our descriptor more compact.

The computation of CBDF includes the following three steps (as illustrated in Fig. 1): (1) Oriented FAST feature point extraction; (2) Oriented binary discriminative feature descriptors; (3) Descriptor compression.

3.1 Oriented FAST Keypoint Extraction

FAST keypoint detector is extensively used in computer vision applications for its efficiency and rotation invariance. There is only one parameter in FAST, which is the intensity threshold between the center pixel and those in a circular ring around the center. Typically 9–16 mask is usually used, it require at least 9 consecutive pixels in the 16-pixel circle which are sufficiently brighter or darker than the central pixel. In such a way, FAST detector has large responses along edges. We defined s as the maximum threshold considering an image point as a corner. The currently testing point needs to fulfill the maximum condition with respect to its 8 neighboring FAST scores. We do not employ a scale pyramid of the image, as for consecutive UAV frames the image scales are almost all the same. Without scale pyramid transformation of input

image, we can also save much time on feature point detection and matching. To measure the corner orientation, we assume that the intensity of corner is offset from its center, as intensity centroid \mathbf{C} named in [16], this vector is used to calculate an orientation. The patch moments are defined as:

$$m_{pq} = \sum_{x,y} x^p y^q I(x, y) \quad (1)$$

Then these moments are used to compute the centroid of a patch as defined in Eq. (2).

$$\mathbf{C} = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right) \quad (2)$$

The orientation of the patch is:

$$\theta = \text{atan2}(m_{01}, m_{10}) \quad (3)$$

where atan2 is the quadrant-aware version of \arctan . To guarantee the rotation invariance, we choose the pixels with positions within a circular area of radius r to calculate the moments.

3.2 Oriented Binary Discriminative Feature Descriptor

Most recent local feature descriptors are based on the statistics of the gradients of pixel intensity in a patch. e.g. SIFT, SURF, GLOH This is because gradients is highly distinctive yet as invariant as possible to remaining variations, such as change in illumination or 3D viewpoint [3]. We follow this trend to build our binary descriptor, yet in a much simplified way. After normalizing the rotation, the proposed CBDF descriptor applies the sampling pattern rotated by θ around the detected keypoints with patch size 32×32 . Then, the intensities of the rotated patch is calculated by nearest neighbor interpolation.

After rotation, gradients of each pixels are computed by a discrete derivative masks. There are several derivative masks can be used to calculate the gradients, such as 1-D point derivatives uncentred $[-1, 1]$, centred $[-1, 0, 1]$, as well as 2×2 diagonal ones $\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$, $\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$, 3×3 Sobel mask and Prewitt mask, we tested these masks in our experiments to chose the best one. And then, gradients are smoothed by Gaussian smoothing. The size of the smoothing template is also tested including $\sigma = 0$ (none). From the experiment results, we find that Sobel mask at $\sigma = 4$ with a size of 5×5 gaussian kernel window works best, and the 1-D point derivative uncentred mask performs almost the same with Sobel, since it's much time saving than Sobel mask, we choose it for our final descriptor. Experiment results are given in Sect. 4. The image patch is then split up regularly into 16 smaller sub-regions in two styles as shown in Fig. 3, and a accumulated gradient magnitude vector \mathbf{v} is calculated for each sub-region:

$$\mathbf{v} = \left(\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y| \right) \quad (4)$$

After this step, each sub region has a four-parameter vector, from which our bit-vector descriptor \mathbf{x} is assembled by a comparison of these parameters between each vector, such that each bit b corresponds to:

$$\forall i = 1, \dots, S, j = 1, \dots, S, k = 1, \dots, 4, i \neq j \quad b = \begin{cases} 1, & \mathbf{v}_{ik} > \mathbf{v}_{jk} \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

where S is the number of the sub regions, k denotes which parameter of the vector to be compared.

For the sake of generating a low bit-rate binary descriptor, we do not compare all sub-regions. Because if we do so, the descriptor length will be $C_{16}^{15} \times 4 = 480$, it is a little too long. Instead, we compare the vector of sub regions linked by the arrow in different and sparse styles as shown in Fig. 2. The radius of the three circles in GV, GVI, GVII, GVIII are set to be 2, 10, 15 pixels. The performance of the different test strategies are given in Sect. 4, experiment results show that even be a 224 bits descriptor, our method out performs several longer descriptors, e.g. BRIEF, ORB, BRISK.

3.3 Descriptor Compression

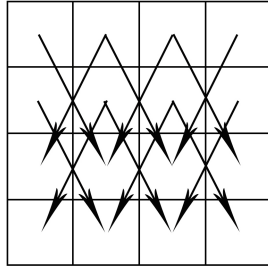
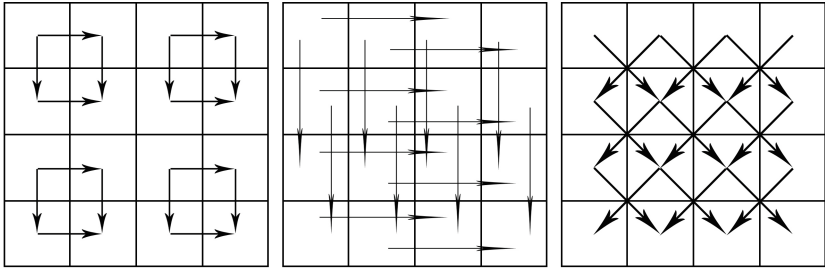
Fewer dimensions mean low memory footprint and fast to match, although our descriptor is much shorter than several state-of-art binary descriptors, we apply a dimension reduction process to our descriptor. However, it is important that we do not adversely affect the performance of the descriptor.

Our keypoint descriptor \mathbf{x} is represented as n -dimensional binary vector in hamming space \mathbf{H}^n . We attempt to find a $m \times n$ matrix \mathbf{P} which takes its value in $\{0, 1\}$ to map our descriptor to an m -dimensional hamming space \mathbf{H}^m . Our goal in finding such a matrix is in two-fold. First, \mathbf{H}^m should be a more efficient representation. This implies that m must be smaller than n . Secondly, through this mapping, the performance should not degrade too much. To better take advantage of training data, we present a supervised optimization scheme that is inspired by [20, 24]. In [24], they use AdaBoost to compute the projection matrix, but there is no guarantee the solution it finds is optimal. We compute a projection matrix that is designed to minimize the in-class covariance of the descriptors and maximize the covariance across classes. In essence, we perform Linear Discriminant Analysis (LDA) on the descriptor.

Here, we limit our attention to dimension reduction of the form:

$$\mathbf{y} = \mathbf{P}\mathbf{x} \quad (6)$$

\mathbf{y} is constructed to minimize the expectation of the Hamming distance on the set of positive pairs, while maximizing it on the set of negative pairs. This can be expressed as minimization of the loss function:

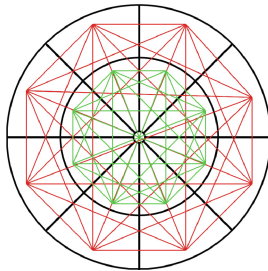
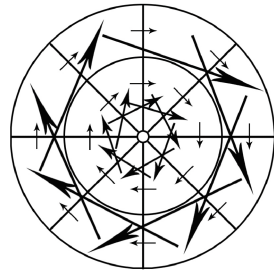
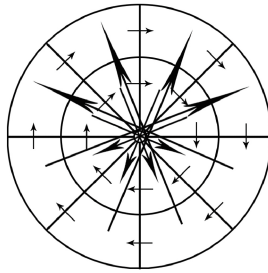
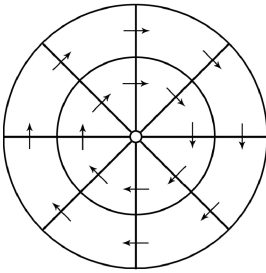


GI

GII

GIII

GIV



GV

GVI

GVII

GVIII

Fig. 2. The binary test strategies, note that we compare the vector parameters of the sub regions linked by the arrows

$$L = E\{d_{\mathbf{H}^m}(\mathbf{y}, \mathbf{y}')|P\} - E\{d_{\mathbf{H}^m}(\mathbf{y}, \mathbf{y}')|N\} \quad (7)$$

Equation (7) is equivalent to the minimization of:

$$L = E\{\|\mathbf{y} - \mathbf{y}'\|^2|P\} - E\{\|\mathbf{y} - \mathbf{y}'\|^2|N\} \quad (8)$$

The direct minimization of Eq. (8) is difficult since the solution of the resulting non-convex problem in $m \times n$ variables is challenging. It can be found that:

$$E\{\|\mathbf{P}\mathbf{x} - \mathbf{P}\mathbf{x}'\|^2|P\} = \text{tr}(\mathbf{P}\Sigma_P\mathbf{P}^T) \quad (9)$$

where $\Sigma_P = E\{(\mathbf{x} - \mathbf{x}')(\mathbf{x} - \mathbf{x}')^T|P\}$ is the covariance matrix of the positive descriptor vector differences, Eq. (9) turns to be:

$$L = \text{tr}\{\mathbf{P}\Sigma_P\mathbf{P}^T\} - \text{tr}\{\mathbf{P}\Sigma_N\mathbf{P}^T\} \quad (10)$$

Pre-multiplying \mathbf{x} by Σ_N^{-1} turns the second term of Eq. (10) into a constant, leaving:

$$L \propto \text{tr}\{\mathbf{P}\Sigma_P\Sigma_N^{-1}\mathbf{P}\} = \text{tr}\{\mathbf{P}\Sigma_R\mathbf{P}^T\} \quad (11)$$

where $\Sigma_R = \Sigma_P\Sigma_N^{-1}$ is the ratio of the positive and negative covariance matrices. Since Σ_R is a symmetric positive semi-definite matrix, it admits the eigendecomposition $\Sigma_R = \mathbf{U}\mathbf{S}\mathbf{U}^T$, where \mathbf{S} is a non-negative diagonal matrix. An orthogonal $m \times n$ matrix \mathbf{P} minimizing $\text{tr}\{\mathbf{P}\Sigma_R\mathbf{P}^T\}$ is a projection onto the space spanned by the m smallest eigenvectors of Σ_R , this yields:

$$\mathbf{P} = \tilde{\mathbf{U}}_m \quad (12)$$

where $\tilde{\mathbf{U}}_m$ is the $m \times n$ matrix with the corresponding eigenvectors. Note that we aim to find a $m \times n$ projection matrix \mathbf{P} which takes its value in $\{0, 1\}$, the result in Eq. (12) does not conform this. The index of the m smallest elements of the principal diagonal elements of Σ_R is denoted as \mathbf{S} . We approximate \mathbf{P} by setting the elements of $\mathbf{P}(\text{ind}, \mathbf{S}_{\text{ind}}) = 1, \text{ind} = 1, \dots, m$ and others to be 0. m is set to be 196 and 128, the original 224 bits U-CBDF descriptor is compressed to 196 bits and 128 bits, denote as CBDF_{196} , CBDF_{128} .

4 Experimental Results

In this section, we first describe our evaluation framework, and then present a set of initial experiments. These experiments validate our approach and allow us to select the appropriate parameters for the descriptor. Finally, we compare our method to other descriptors including BRIEF, ORB, BRISK, SIFT and SURF. Finally, we apply our proposed CBDF descriptor in a real UAV image registration application.

4.1 Performance Evaluation Protocol

We evaluate the performance of our method using two datasets. The first dataset is proposed by Mikolajczyk and Schmid [6, 11]. This dataset contains several sub datasets. Each of the sub datasets contains a sequence of six images exhibiting an increasing amount of transformation. This dataset is used to detect the appropriate parameters for our descriptor. We use precision rate as a quality criterion, we show the Nb best matches and count the number of correct matches n_c . The precision rate is calculated by $r = n_c / Nb$. We set $Nb = 300$ in our experiment, we tune the threshold of each method to get 300 best matches. However, it's usually hard to get exactly 300 matches, we get an approximate number, and the deviation is constrained no more than 2.

The second dataset contains two sub datasets: *Notre Dame* and *Liberty* [14]. Each of them contains over 400 k scale- and rotation-normalized 64×64 patches. These patches are sampled around interest points which detected using Difference of Gaussian, and the correspondences between patches are found using a multi-view stereo algorithm. The resulting datasets exhibit substantial perspective distortion and light changing conditions. The ground truth available for each of these datasets describes 100k, 200k and 500k pairs of patches. We train matrix \mathbf{P} with these datasets. The performance of our CBDF descriptor is compared against U-CBDF and the state-of-the-art descriptors. The test set contains 100,000 pairs in which 50 % match pairs, and 50 % non-match pairs.

4.2 Initial Experiments

There are several parameters that influence the performance of our descriptor as been mentioned in Sect. 3: the smoothing scales σ , the size of the smoothing template, the mask to compute the gradient, and the test strategy to generate our descriptor. We use the *Wall* dataset proposed in [6, 11] to test these parameters. It contains five image pairs, with the first image being the same in all pairs and the second image shot from a monotonically growing baseline, which makes matching increasingly more difficult. Figure 4(a) shows the first image of the *Wall* sequence. All the initial experiments are tested on the U-CBDF descriptor. When we test the influence of one of the parameters, other parameters are set to be the correct value which we finally use. Figure 3(a) shows the results obtained for different values of σ . For most of the values of σ , the performance are optimal for $\sigma = 2$, so we keep $\sigma = 2$ in the remaining experiments. Figure 3(b) shows the precision rates for different smoothing templates. The 5×5 mask outperforms other masks, so we keep 5×5 Gaussian smoothing template for our final descriptor. Figure 3(c) shows the influence of different gradient masks, we find that the Sobel mask performs slightly better than the 1-D point derivative uncentred mask, since 1-D point derivative uncentred mask is much time saving than Sobel mask, we choose it for our final descriptor. The influence of different test strategies is shown in Fig. 3(d). We also calculate the precision rate of BRIEF₅₁₂ which has a length of 512 bits. Clearly, the symmetrical and regular GV, GVI, GVII, GVII strategies enjoy a big advantage over the other four in most cases. GVIII performs the best and it has a length of 224 bits. For this reason, in all further experiments presented in this paper, it is the

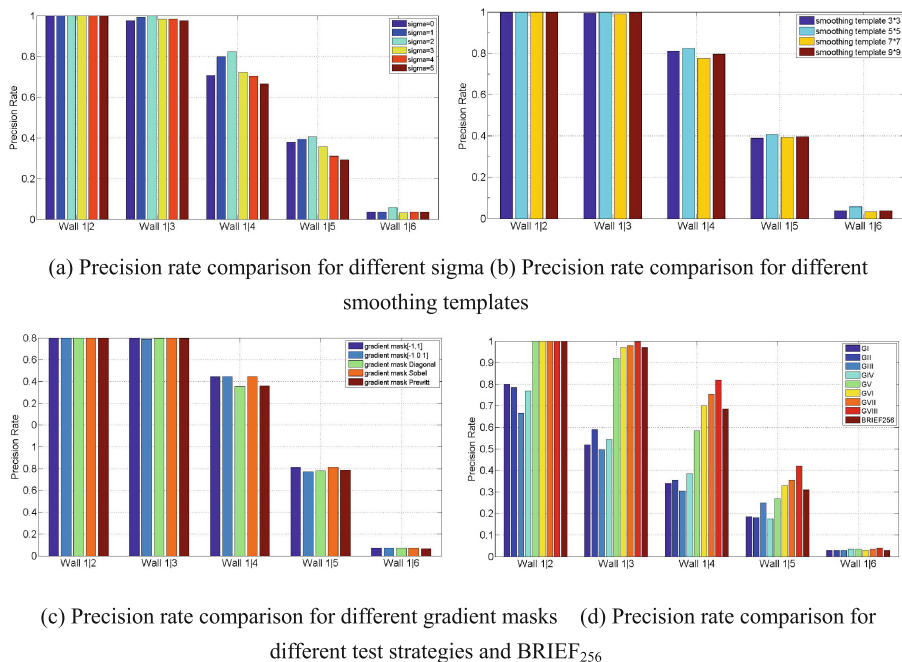


Fig. 3. Precision rate comparison for different parameters and BRIEF₂₅₆

one we will use. We also find that GVII and GVIII strategies perform better than BRIEF₂₅₆ in all cases, in which GVII has a length of only 128 bits.

Using the above-mentioned parameters for our U-CBDF descriptor, we train the matrix \mathbf{P} with both the *Notre Dame* and *Liberty* datasets. \mathbf{P} is used to compress U-CBDF descriptor.

4.3 Descriptor Comparison

In this section, we use the *Notre Dame* and *Liberty* datasets as our training and test datasets. Figure 4(b) shows some image patches from the *Liberty* dataset. We compare our binary descriptors both uncompressed and compressed to the very recent BRIEF, ORB, BRISK binary descriptors, results obtained with SIFT and SURF are also presented. All the experiments are performed on a desktop computer with an Intel core2 2.80 Hz CPU. For SIFT, BRIEF, ORB, and BRISK, we use the publicly available library OpenCV2.4.3. For SURF, we use the implementation available from their authors. During testing, we compute the distances of all match/non-match descriptors, and sweep a threshold on the descriptor distance to generate a ROC curve. We also report 85 % error rate in Table 2, 85 % error rate is the percent of incorrect matches obtained when 85 % of the true matches are found.

Figure 5 provides the ROC curves for U-CBDF, CBDF and the state-of-the-art methods on different training and test datasets. Both Fig. 5(a) and (b) show that

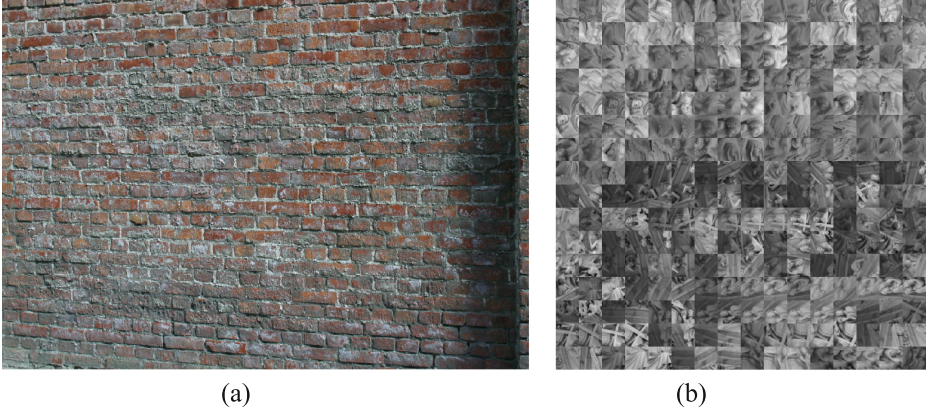


Fig. 4. (a) The first image of *Wall* sequence of the Mikolajczyk and Schmid dataset (b) some image patches from the *Liberty* dataset.

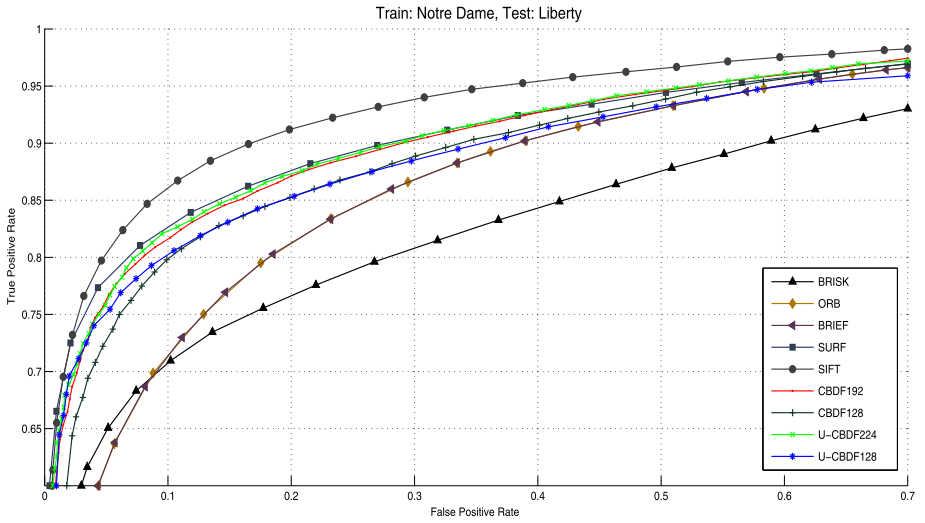
although CBDF_{192} is 32 bits shorter than U-CBDF_{224} , its performance does not degrade too much. CBDF_{192} performs better than its binary competitors at all error rates. CBDF_{192} remains competitive to SURF, even though it has a much shorter representation. SIFT performs the best of all tested descriptors, though its complexity is prohibitive for real-time application. BRISK performs the worst at high false positive rate although it is much longer.

The first row of Table 1 clearly shows that CBDF_{192} provides up to 28 % improvement over BRISK and up to 11 % improvement over BRIEF and ORB in terms of 85 % error rate, CBDF_{128} provides up to 24 % improvement over BRISK and up to 7 % improvement over BRIEF and ORB. While CBDF_{128} requiring only 16 bytes instead of 64 bytes for BRISK and 32 bytes for BRIEF. It also shows that CBDF_{128} remains competitive to the much longer and much more computationally expensive floating-point SURF. The second row of Table 2 shows the similar results with the first row.

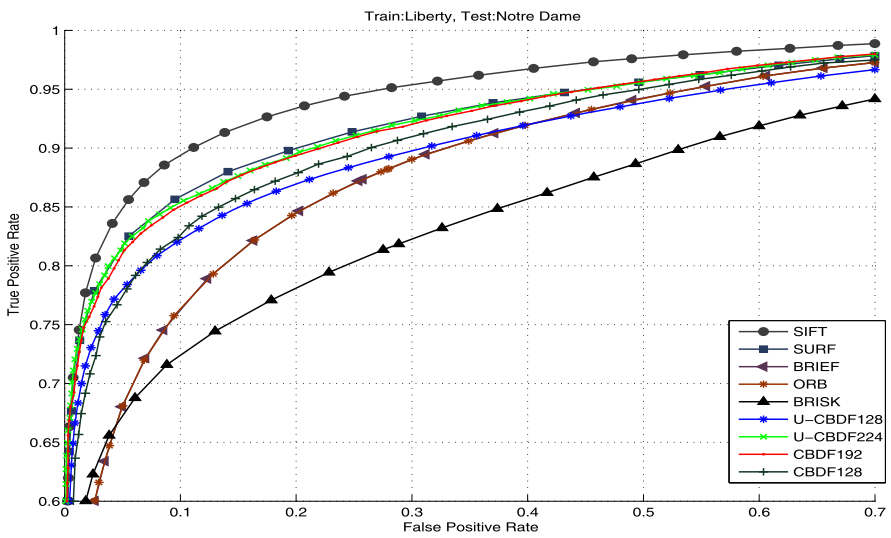
4.4 Timings

The timings of our method and its competitors are extensively tested with the boat image sequence by Mikolajczyk and Schmid dataset which are shown in Table 2. We use the first image and the second image of this sequence (size: 850×680). We also use FAST corner detector for BRIEF just like ORB, because there is no special key-point extractor for BRIEF, so the timings of detection are almost the same. Their differences are in feature descriptor. By tuning the threshold of each method, we extract 1000 keypoints on each image. The matching of each method is based on a brute-force descriptor distance computation. We ran each method for 100 times and calculate the average time cost.

The timings show an advantage of CBDF_{192} . Its descriptor computation time is typically two times faster than the one of SURF, and three times faster than the one of



(a)



(b)

Fig. 5. Comparison of our CBDF descriptor to the state-of-the-art binary and floating-point descriptors

Table 1. 85 % error rate for our CBDF descriptor and BRIEF, BRISK, SURF, SIFT. We give the results for U-CBDF128 and U-CBDF224 which are corresponding to the GVII and GVIII in Fig. 1 and CBDF128, CBDF192 which are compressions of U-CBDF224.

Dataset	SIFT	SURF	BRIEF ₅₁₂	ORB ₅₁₂	BRISK ₅₁₂	U-CBDF ₁₂₈	U-CBDF ₂₂₄	CBDF ₁₂₈	CBDF ₁₉₂
Notre dame	5.04	8.60	20.83	20.83	37.93	14.68	9.15	13.30	9.92
Liberty	8.67	14.08	26.33	26.33	42.25	18.75	15.04	17.87	16.10

Table 2. Timings for different methods

	Timings of feature detection and description				Matching timings	
	Detection [ms]	Description [ms]	Total time [ms]	Time per point [ms]	Matching [ms]	Time per point [ns]
SIFT _{128f}	1687.47	1236.21	2923.68	1.462	1223.99	1223.99
SURF _{64f}	3025.04	826.80	3851.84	1.926	77.97	77.97
BRIEF _{512b}	84.22	16.25	100.47	0.050	61.34	61.34
ORB _{512b}	84.97	55.03	140.00	0.070	61.01	61.01
BRISK _{512b}	81.11	20.33	101.44	0.051	99.60	99.60
CBDF _{128b}	84.36	377.14	461.50	0.231	16.48	16.48
CBDF _{192b}	84.36	377.14	461.50	0.231	23.84	23.84

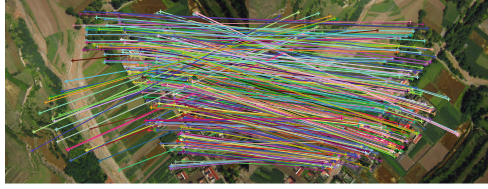
SIFT. The matching timings per point is faster than the one of SIFT, SURF, BRIEF, ORB and BRISK.

4.5 UAV Image Registration

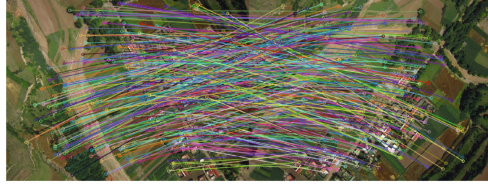
To demonstrate the performance of our proposed CBDF feature descriptor on real UAV image registration, we test the performance of our descriptor on 100 pairs of real UAV images. Figure 6 shows one pair of these images. The image size is 533×400 pixels. There is a large rotation between this pair of images. Both CBDF₁₉₂, BRIEF, ORB, BRISK, SURF and SIFT are tested on these images. The registration results are shown



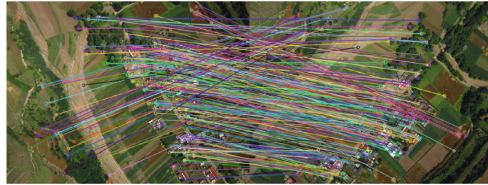
Fig. 6. UAV images to be registration



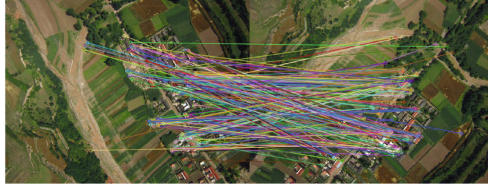
(a)CBDF192 registration results (300 pair points, 100% precision rate)



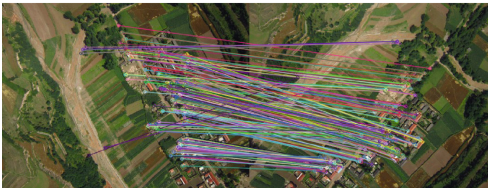
(b)SURF registration results (297 pair points, 100% precision rate)



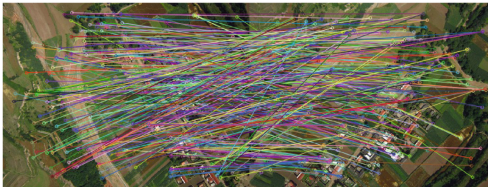
(c) SIFT registration results (295 pair points, 100% precision rate)



(d)BRIEF registration results (304 pair points, 14.47% precision rate)



(e)ORB registration results (312 pair points, 100% precision rate)



(f)BRISK registration results (302 pair points, 62.91% precision rate)

Fig. 7. UAV image registration results

Table 3. Timings for different methods with UAV images

	Points in the two images	Match points	Detection time [ms]	Description time [ms]	Matching time [ms]	Total time [ms]	Inliers rate [%]
CBDF ₁₉₂	1011, 1031	300	34.33	268.70	23.21	326.2	100
SURF	1004, 1001	297	2693.31	660.33	73.20	3426.8	100
SIFT	1000, 1000	295	515.72	734.31	122.58	1372.6	100
BRIEF	1000, 1000	304	35.67	11.53	58.33	105.5	14.47
ORB	1000, 1000	312	34.44	27.22	57.80	119.5	100
BRISK	993, 996	302	121.70	14.97	99.70	236.4	62.91

in Fig. 7. The timings are listed in Table 3. One shall note that all our method is almost 10 times faster than SURF and 4 times faster than SIFT.

5 Conclusions

In this paper, we have defined a new oriented binary discriminative feature for UAV image registration. With only 192 bits or even 128 bits per descriptor, CBDF outperforms its binary state-of-the-art competitors in terms of accuracy while significantly reducing the memory footprint, and comparing to SIFT and SURF, the method offers faster alternative at comparable matching performance. Experiments with real UAV images justify that this framework can fulfill the near real-time image registration application requirement.

Acknowledgments. This work is supported by the Nature Science Foundation of China (No.11373043), the National 863 Project of China (No.2014****), and the Collaborative Innovation Special Foundation of Xuchang University.

References

1. Zhou, G.: Geo-referencing of video flow from small low-cost civilian UAV. *IEEE Trans. Autom. Sci. Eng.* 7(1), 156–166 (2010)
2. Lowe, D.G.: Object recognition from local scale-invariant features. In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, pp. 1150–1157 (1999)
3. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* 60(2), 91–110 (2004)
4. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: speeded up robust features. In: *Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006, Part I. LNCS*, vol. 3951, pp. 404–417. Springer, Heidelberg (2006)
5. Ke, Y., Sukthankar, R.: PCA-SIFT: a more distinctive representation for local image descriptors. In: *The 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, p. II-506 (2004)

6. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.* **27**, 1615–1630 (2005)
7. Calonder, M., Lepetit, V., Strecha, C., Fua, P.: BRIEF: binary robust independent elementary features. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) *ECCV 2010, Part IV. LNCS*, vol. 6314, pp. 778–792. Springer, Heidelberg (2010)
8. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: ORB: an efficient alternative to SIFT or SURF. In: 2011 IEEE International Conference on Computer Vision (ICCV), pp. 2564–2571 (2011)
9. Leutenegger, S., Chli, M., Siegwart, R.Y.: BRISK: binary robust invariant scalable keypoints. In: 2011 IEEE International Conference on Computer Vision (ICCV), pp. 2548–2555 (2011)
10. Alahi, A., Ortiz, R., Vandergheynst, P.: Freak: fast retina keypoint. In: The 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 510–517 (2012)
11. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Van Gool, L.: A comparison of affine region detectors. *Int. J. Comput. Vis.* **65**(1–2), 43–72 (2005)
12. Gionis, A., Indyk, P., Motwani, R.: Similarity search in high dimensions via hashing. In: *Proceedings of The International Conference on Very Large Data Bases*, pp. 518–529 (1999)
13. Weiss, Y., Torralba, A., Fergus, R.: Spectral hashing. In: *NIPS* (2008)
14. Gong, Y., Lazebnik, S.: Iterative quantization: a procrustean approach to learning binary codes. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 817–824 (2011)
15. Rosten, E., Drummond, T.W.: Machine learning for high-speed corner detection. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) *ECCV 2006, Part I. LNCS*, vol. 3951, pp. 430–443. Springer, Heidelberg (2006)
16. Rosin, P.L.: Measuring corner properties. *Comput. Vis. Image Underst.* **73**(2), 291–307 (1999)
17. Harris, C., Stephens, M.: A combined corner and edge detector. In: *Alvey Vision Conference*, vol. 15, pp. 50 (1988)
18. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide-baseline stereo from maximally stable extremal regions. *Image Vis. Comput.* **22**(10), 761–767 (2004)
19. Mikolajczyk, K., Schmid, C.: Scale & affine invariant interest point detectors. *Int. J. Comput. Vis.* **60**(1), 63–86 (2004)
20. Strecha, C., Bronstein, A.M., Bronstein, M.M., Fua, P.: LDAHash: improved matching with smaller descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.* **34**(1), 66–78 (2012)
21. Chandrasekhar, V., Takacs, G., Chen, D., Tsai, S., Grzeszczuk, R., Girod, B.: CHoG: compressed histogram of gradients a low bit-rate feature descriptor. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2504–2511 (2009)
22. Brown, M., Hua, G., Winder, S.: Discriminative learning of local image descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.* **33**(1), 43–57 (2011)
23. Tola, E., Lepetit, V., Fua, P.: Daisy: An efficient dense descriptor applied to wide-baseline stereo. *IEEE Trans. Pattern Anal. Mach. Intell.* **32**(5), 815–830 (2010)
24. Shakhnarovich, G.: Learning task-specific similarity. Ph.D. dissertation, MIT (2005)