Training Binary Descriptors for Improved Robustness and Efficiency in Real-Time Matching

Sharat Saurabh Akhoury and Robert Laganière

School of Electrical Engineering and Computer Science,
University of Ottawa, Ottawa, Ontario, Canada
{sakhoury,laganier}@uottawa.ca

Abstract. Most descriptor-based keypoint recognition methods require computationally expensive patch preprocessing to obtain insensitivity to various kinds of deformations. This limits their applicability towards real-time applications on low-powered devices such as mobile phones. In this paper, we focus on descriptors which are relatively weak (i.e. sensitive to scale and rotation), and present a classification-based approach to improve their robustness and efficiency to achieve real-time matching. We demonstrate our method by applying it to BRIEF [7] resulting in comparable robustness to SIFT [4], while outperforming several state-of-the-art descriptors like SURF [6], ORB [8], and FREAK [10].

Keywords: keypoint recognition, feature matching, object detection.

1 Introduction

Computer vision applications, including object recognition, image retrieval, visual odometry and Augmented Reality (AR), heavily rely on establishing correspondences between similar real world points appearing in various images. The most commonly accepted and widely used method for achieving this is based on keypoint recognition, which comprises of two fundamental stages. The first stage involves detecting keypoints in an image containing the object of interest. The second stage comprises of feature extraction, whereby a feature is computed to describe the region encompassing the keypoint. Correspondences can then be found by matching the features from a source and query image.

Both stages have received a considerable amount of attention over the last few decades, with recent algorithms being designed to be more robust against various deformations while minimizing the processing time. State-of-the-art keypoint detectors such as FAST [1], CenSurE [2], and AGAST [3] have been shown to not only reliably detect keypoints, but perform exceptionally well on low-powered devices [14, 15]. A variety of features derived from the local image intensities have been proposed to yield robust descriptors with the most seminal contributions coming from [4, 12, 13]. In general, feature descriptors can be categorized into two broad classes: *descriptor-based* and *classification-based* [15].

A. Petrosino (Ed.): ICIAP 2013, Part II, LNCS 8157, pp. 288–298, 2013.

Descriptor-based approaches rely on designing the feature to be as robust as possible against specific types of deformations. They typically require fine scale selection, rotation correction, and intensity normalization. Some well-known descriptor-based approaches include: SIFT [4], GLOH [5], and SURF [6]. The large computational requirements of these descriptors generally limit their applicability towards real-time applications. Recent attempts have been made to decrease the processing time: ORB [8], BRISK [9], and FREAK [10].

Classification-based approaches, by contrast, require an offline training stage to learn feature sets from synthesized images (of the source object) allowing us to achieve insensitivity to specific kinds of deformations. These approaches are designed to transfer much of the computational burden to the training phase in order to reduce the cost of online matching while increasing its robustness. Prior work [12–15] that have been done on classification-based descriptors still face drawbacks in terms of high training times and/or memory usage.

Some descriptor-based approaches, such as BRIEF [7], are designed to be very fast to compute but are robust to a limited range of deformations only. In this paper, we show how to extend the robustness of such descriptors by adopting a classification-based approach. We present a model generation framework which can be applied to any descriptor-based approach, allowing us to achieve real-time matching. We strictly focus on objects which are planar, thus allowing us to synthesize images of the object in order to capture the appearance of keypoint patches under several perspectives [11–14].

This paper is structured as follows: In Sect. 2, we review the literature on classification-based approaches and summarize our contributions. Section 3 presents the generic training framework. In Sect. 4, we discuss a novel concept on verifying a feature's stability. Section 5 presents our classification approach. In Sect. 6 we describe three variants of the modified BRIEF descriptor using our proposed training framework. Section 7 details the experimental setup and presents the datasets that were collected for evaluation. Section 8 presents our results obtained. Finally, Sect. 9 presents our conclusions.

2 Related Work

Lepetit et al. [11] revolutionized the keypoint matching problem by casting it as a classification problem. Given an image of a target object, a feature set $\hat{F}$ is generated by combining the statistics of the set of warped patches, referred to as a viewset, around each keypoint. Each detected keypoint denotes a distinct class. Invariance to various deformations can be achieved by synthesizing a sufficient amount of viewsets. PCA was applied in [11] to extract a compact description for all the viewsets associated with each keypoint. At runtime, a nearest-neighbour (NN) classifier was then employed to identify keypoints which are in $\hat{F}$.

Lepetit and Fua [12] further extended their initial approach in [11] by replacing the NN classifier with a random forest. Image patches were recognized on the basis of very simple, randomly chosen binary tests which were grouped into decision trees which then recursively partitioned the space of all possible viewsets.

The random forest classifier was later replaced by the naive Bayesian classifier resulting in the well known Fern descriptor [13]. A major limitation to Fern is that it requires large amounts of memory in order to represent the complicated joint distributions for each feature for the Bayesian classifier.

Taylor et al. [14] presented another training-based keypoint recognition approach, which generates a feature set by computing coarse histograms of the intensities of selected pixels around a keypoint for each viewset. They refer to their method as Histogrammed Intensity Patches (HIP). Although the training process is quite similar to that used in [12, 13], a slight modification is made pertaining to the viewset generation process. For each keypoint, instead of clustering all the viewsets into a single class, Taylor et al. [14] proposed organizing them into separate viewpoint bins such that each each bin is characterized by a similar set of deformations. Although fairly good matching robustness can be achieved, the training method is computationally expensive as the authors report that it takes at least 20 minutes to generate a model [14, 15].

2.1 Our Contributions

We have developed a generic framework which inherits from the classification-based paradigm, in order to improve the robustness and efficiency of relatively weak descriptor-based approaches. We integrate the key principles from [12–14], and achieve more than 80% reduction in training time as compared to [14, 15]. Our reduction comes from the way we synthesize the viewset images. Instead of applying the conventional affine transformation, we apply a perspective transformation which allows us to sample the deformation space in a more controlled way requiring less images per viewset. Furthermore, we also present a novel concept of verifying a feature's stability in order to generate a reliable model. Lastly, using our framework, we propose three variants to the BRIEF descriptor and demonstrate comparable performance to SIFT.

There are two factors which ultimately determine the system performance, namely the repeatability of the keypoint detector and the "descriptiveness" of the descriptor. Ideally, it is desired to have a fast keypoint detector, however, such a detector generally compromises speed for repeatability. If similar real-world points cannot be consistently detected, then even though we might have extracted features for those points in our model database, those features may never be matched. The descriptiveness of the descriptor is a significant contributory factor which directly influences the overall success of our proposed framework. Although our system is designed to increase the robustness of (relatively weak) descriptors to various deformations such as rotation, scale and perspective distortions; using an arbitrarily designed descriptor which does not uniquely describe the local neighborhood of the keypoint results in more false matches. Such a non-representative descriptor yields higher outlier ratios which results in more processing time for estimating the pose.

3 Model Generation Framework

Constructing a model, $\hat{F}$, comprises of three primary steps: viewset generation, keypoint detection & stable point identification, and feature extraction & aggregation. These steps are performed in sequential order.

3.1 Viewset Generation

Multiple artificial views of the target image $\hat{I}$, are synthesized and grouped into numerous viewpoint bins so that each bin shares a similar range of deformations. To avoid the use of a priori information of camera intrinsics, most approaches [11–15] apply the affine transformation to synthesize these images. We step away from this convention and apply a perspective transformation instead. Since our target application is for cameras on low powered devices such as mobile phones, it is possible to use approximate camera calibration parameters that match well with real world scenarios. This allows us to sample the deformation space in a more representative, controlled way requiring less images per viewset to cover a specified range of deformations. We therefore fixed the vertical field of view of the camera at 30° and set our virtual plane at the distance such that the view frustum has a width equal to the plane diagonal. This ensures that the plane will remain entirely visible under an arbitrary rotation around any of the three axes. We specify the viewpoint bin range in terms of the rotation angles around all three axes, $[\Phi^x_{min}, \Phi^x_{max}]$, $[\Phi^y_{min}, \Phi^y_{max}]$,$[\Phi^z_{min}, \Phi^z_{max}]$ and the scale, $[S_{min}, S_{max}]$. In a further attempt to simulate real world images, we also add Gaussian noise after convoluting each synthesized image I with a Gaussian kernel modeling the optical blur:

$$I = \mathcal{G} * \mathcal{V}\left(\hat{I}\right) + \mathcal{N}(0, \sigma) \;, \tag{1}$$

where $\mathcal{V}()$ denotes the perspective transformation process, $\mathcal{G}$ is the Gaussian convolution kernel, and $\mathcal{N}(0, \sigma)$ denotes the additive Gaussian noise with mean 0 and standard deviation σ. We commonly refer to $\hat{I}$ as the reference image in this paper. Each viewpoint bin comprises of M images synthesized using Eq. (1) with random angle and scale distortions selected from the viewpoint bin range. Each bin range is determined by a step $\Delta\Phi$ computed as:

$$\Delta\Phi^x = \frac{\Phi^x_{max} - \Phi^x_{min}}{B_x} \;, \tag{2}$$

where B_x is the number of bins along the x axis. The angle range for bin b is specified by $\Phi^x_{min} + b\Delta\Phi^x \leq \phi_x < \Phi^x_{min} + (b+1)\Delta\Phi^x$. Similar equations hold for the two other axes. Binning of the scale follows a multiplicative scheme:

$$\gamma = \sqrt[B_s - 1]{\frac{S_{min}}{S_{max}}} \;, \tag{3}$$

with bin b including scales in the range $\gamma^{b+1} S_{max} \leq s \leq \gamma^b S_{max}$. When combined together, the total number of viewpoint bins is $N = B_x B_y B_z B_s$ which gives a total of $M \times N$ virtual views to be generated. We denote the m^{th} virtual image from the n^{th} viewpoint bin as I_{mn}.

3.2 Keypoint Detection and Stable Point Identification

Keypoints are detected from each viewpoint image (for the viewpoint bin under consideration) using a fast interest point detector such as [1–3]. Before extracting the descriptors for each keypoint, we first determine the Q most stable points. A stable point comprises of two fundamental components: a point $\hat{p}_r$ on the reference image, and a group of keypoints, G_{nr}, from the n^{th} viewpoint bin which map back to $\hat{p}_r$. Stability can be measured by the repeatability of keypoints detected across various viewpoints in the bin. This is done as follows. The keypoints from I_{mn} are converted to a reference position on $\hat{I}$ by computing $\mathcal{V}_{mn}\left(\hat{I}\right)^{-1}$, which is the inverse mapping of Eq. (1). We then find groups of keypoints, G_{nr}, which approximately map back to the same reference position $\hat{p}_r$. The number of candidate stable points is therefore specified by the size of G_{nr}. The stable points are then ranked according to the corresponding group size $|G_{nr}| \leq M$. The Q highly ranked stable points are then selected from G_{nr}, as these represent the set of most stable keypoints for the n^{th} viewpoint bin.

This strategy works exceptionally well for relatively large values of Q, where we are allowed a large budget to select many stable points. However, for smaller Q, we can run the risk of only selecting stable points from a dense high-scoring region of the image, increasing the system's vulnerability to occlusion. Hence, we adopt a stratified sampling technique to select the stable points across the entire image thereby improving the system's robustness. There are two possible ways of performing stratified sampling. The first way is to select stable points local to the viewpoint bin under consideration. The second way is to pre-determine the global stable points using a separate set of synthesized images representing a broad spectrum of random deformations as in [12, 13]. In both cases, we tessellate the reference image into $W \times W$ subwindows and select the Q/W^2 highly ranked stable points from each subwindow. In our experiments, we found that the first strategy yielded in significantly better matching performances.

3.3 Feature Extraction and Aggregation

After the stable points have been identified, we then extract features for all the keypoints associated with each stable point. We refer to the individual features as partial features. The final feature which describes the stable point is obtained by aggregating all the corresponding partial features. For most binary descriptors, aggregation can be performed by a simple majority vote on each bit of the descriptor set. For other binary descriptors which make use of histograms (such as the HIP descriptor [14]), instead of a majority vote, a cut-off threshold can be applied to identify rarely occupied bins. Floating-point descriptors can be aggregated by computing the average of each feature dimension.

4 Feature Stability Verification

Being able to identify keypoints which are detectable across various deformations does not necessarily imply that the associated features are reliable. Since we

aggregate multiple partial features to obtain the final feature, a situation can arise where the viewsets contain too much variation in appearances, resulting in an unreliable, noisy feature. We propose computing a supplementary feature vector, which we call the stability vector, Ω, in order to identify and reject such random features.

For binary feature descriptors, Ω is determined as follows. For each stable keypoint, we compute the entropy of each bit of the binary representation associated with the group of keypoints from the different virtual views. This entropy measure is based on the fraction of times a given bit has a value 1, which is simply $P(1)$, and similarly for $P(0)$.

$$Entropy(G_{nr}) = -P(1)\log P(1) - P(0)\log P(0)\ . \tag{4}$$

To be considered stable, the entropy of a bit must be lower than a given threshold and the corresponding bit of Ω is set to 1. After all the bits of the feature have been processed, we perform a bitcount on Ω and compare it to a threshold Ω_s (the stability threshold). If the number of stable bits is less than this threshold, then the representation is considered unstable, and is rejected.

Furthermore, by storing the stability vector associated with each feature in $\hat{F}$, we can exploit the knowledge of which elements of the feature are stable for establishing reliable correspondences. Thus, when matching features, we only compare the bits (for binary features) or dimensions (for floating point features) that were deemed stable. This comes at the cost of doubling the descriptor size.

5 Keypoint Classification and Recognition

After generating the model $\hat{F}$, a classifier can be trained to learn the feature space in order to recognize keypoints from query images. In this paper, we avoid an additional classifier training stage by employing the nearest-neighbour classifier using a locality sensitive hashing (LSH) approach [16]. By representing $\hat{F}$ in a compact binary representation comprising of hash values, we are able to efficiently match features in real-time. We set our hash function according to the 13-bit indexing scheme used in [14].

In situations where the query images are severely distorted by motion blur, keypoint detection algorithms can fail to locate the query keypoints which are also in $\hat{F}$. Instead of applying a computationally expensive scale-invariant keypoint detector, we advocate the use of downsampled query images to create a pyramid of η levels. For efficient runtime performances, we downsample by a factor of 2. Our experiments indicate that $\eta \leq 3$ yields sufficiently good localization results. An added bonus of the pyramid is that it also facilitates the detection of targets at multiple scales.

6 Modified BRIEF Feature Descriptor

BRIEF [7] is a recent feature descriptor, which, similar to [12, 13], makes use of simple binary tests based on intensity differences between pixels in a smoothed

image patch. The descriptor comprises of an n-bit string description of an image patch constructed from a set of binary intensity tests. Calonder et al. [7] have demonstrated the BRIEF descriptor to be highly discriminative and invariant to changes in illumination, blur as well as partial perspective distortions. However, BRIEF remains sensitive to both scale and rotation deformations. Hence, it is a perfect candidate to be applied to our framework. We propose here three variants to the BRIEF descriptor by extending the original algorithm using our training framework, namely *cBRIEF*, *sBRIEF*, and *s+BRIEF*. They all operate similarly, however, the latter two make use of the feature stability verification process during training (sBRIEF) and matching (s+BRIEF). In this paper, we set the descriptor length to 256 bits, and generated the binary test locations according to the optimal distribution found in [7] for a 31×31 patch.

7 Experimental Setup

In addition to the three modified BRIEF descriptors, we also implemented the HIP feature descriptor as described in [14] using our model generation framework. The HIP descriptor serves as a benchmark to compare against our BRIEF descriptors, since Taylor et al. [14] have shown HIP to have similar robustness to SIFT [4] and Ferns [13]. We applied the FAST detector [1] for keypoint detection and RANSAC for pose estimation. All the experiments in this paper were performed on a 2.67 GHz processor running Ubuntu 12.04 (32-bit).

7.1 Dataset

We captured five sets of image sequences for five different types of targets (shown in Fig. 1), with each sequence comprising of around 250 to 300 frames. The image sequences were captured using an LG Optimus 2X smartphone camera with a resolution of 480×480. The camera was rotated by approximately $45°$ in all directions (i.e. $45°$ in- and out-of-plane rotation). The scale of the target varied from full resolution (where the target fully occupies the frame) to one third resolution (where the target occupies no more than a third of the frame). A majority of the images suffer from further perspective distortions and severe motion blur in some cases. The ground truth target locations were manually obtained by identifying the four corners of the target in each image sequence.

7.2 Evaluation Metrics

For low-powered devices such as mobile phones, the overall system performance in terms of accuracy, speed and memory consumption is of greatest concern. The accuracy relates the ability to detect a target, whereas the precision (which specifies the inlier ratio) provides important information with regard to the "descriptiveness" of the descriptor. Furthermore, it is the precision metric which directly influences the time required to estimate the pose. To this end, the following performance metrics were used in this paper: *accuracy*, *precision*, *model size* ($|\hat{F}|$), *file size* (F_s), and *training time*.

Fig. 1. 5 frames of the test sequences (from left to right): advertisement, book cover, football photograph, printed map, and text-based image

The accuracy of a detection is determined by analyzing the maximum error between the estimated target corner locations to the corresponding ground truth corner location. The maximum error is obtained as follows:

$$\mathcal{E}(\tilde{C}) = \max_{j} \|C_j - \tilde{C}_j\|,\ 1 \leq j \leq 4\ , \tag{5}$$

where C_j and $\tilde{C}_j$ denotes the ground truth and estimated location of corner j. The four corners are arranged in the following manner: top left (C_1), top right (C_2), bottom right (C_3), and bottom left (C_4). The target is successfully detected if $\mathcal{E}(\tilde{C}) \leq \delta$. We set δ to 10 to allow for errors up-to 10 pixels in target localisation to account for severely blurred images in the datasets. In these cases, it was nearly impossible to precisely specify the ground truth corners. The precision metric represents the inlier ratio which is computed as the fraction of correct matches over the total number of matched points. We allow an error of up-to 5 pixels in identifying correct matches.

It is important to note that the choice of the accuracy and precision error thresholds (currently chosen to be 10 and 5 pixels) are arbitrary in the sense that they do not influence the relationship between the various descriptors evaluated. Setting these thresholds to very high values provide too optimistic results, on the contrary, stricter thresholds give rise to over pessimistic results. Nonetheless, the ranking between the descriptors remains the same.

The model size metric specifies the average number of features contained in the model, whereas the file size metric relates the actual size of the model on disk. We store the x and y reference position of a feature using two 16-bit unsigned integers. The file size (in bytes) is thus calculated as follows:

$$F_s = |\hat{F}| \times (4 + \lambda)\ , \tag{6}$$

where λ is the descriptor size in bytes. The training time metric quantifies the time taken to generate the model and is measured in minutes.

8 Results

In order to establish the optimal set of values to use for the model generation framework, we have conducted an extensive empirical investigation. We set Φ_x and Φ_y to range from $[-30°, 30°]$ with $\Delta\Phi^x = \Delta\Phi^y = 30°$, and Φ_z to range from $[-50°, 50°]$ with $\Delta\Phi^z = \{5°, 10°, 15°, 20°\}$. The scale parameter was configured to the range $S \in [\frac{1}{3}, 1]\times$ resolution of the reference image, with $\gamma = \{0.7, 0.8, 0.9\}$. The tested values for the other parameters are

Table 1. Evaluation results for HIP, cBRIEF, sBRIEF and s+BRIEF obtained with the optimal configuration

Descriptor	λ (bytes)	Accuracy	Precision	$\lvert\hat{F}\rvert$	F_s	Time (min)
HIP	40	84.82%	38.43%	11936	513 KB	1.29
cBRIEF	32	84.93%	53.20%	11936	420 KB	2.00
sBRIEF	32	84.88%	53.22%	11746	413 KB	2.14
s+BRIEF	64	84.34%	61.33%	11746	780 KB	2.16

$S_{max} = \{600, 400, 200, 100\}$, $M = \{50, 100, 200\}$, $Q = \{50, 75, 100\}$, $W = \{1, 2, 3, 4\}$ and $\Omega_S = \{128, 192\}$. In search for the most accurate configuration for each descriptor (i.e. the one with best target detection rate), we exhaustively tested all possible combinations of parameters on our test video sequences, resulting in 1728 configurations (twice this number for sBRIEF and s+BRIEF). The overall best configuration was found to be $\Delta\Phi^z = 20°$, $M = 100$, $Q = 75$, $W = 4$, and $\Omega_s = 128$. With regard to the scale, we observed that $\gamma = 0.9$ produced optimal results, however, at the cost of relatively large models since 12 scale bins are required to cover the range $S \in [\frac{1}{3}, 1]$. We observed a similar performance (i.e. a drop of less than 1% in accuracy) by using $\gamma = 0.85$ (which requires 8 scale bins). The evaluation results of all 4 descriptors using the optimal configuration are presented in Table 1.

Note that we are able to match HIP in accuracy while obtaining a significantly better precision (greater than 15%). cBRIEF and sBRIEF perform quite similarly, with a slight advantage provided by sBRIEF in terms of a smaller model and file size (as evident in Table 1). The advantage of applying the feature stability verification process, during model generation and runtime matching, is demonstrated by the s+BRIEF descriptor. For the same configuration, s+BRIEF boasts of a gain in precision of up to 8% greater than cBRIEF and sBRIEF. However, the gain comes at the cost of an increased file size of the model database, which is almost twice than that of cBRIEF and sBRIEF.

We also evaluated our improved BRIEF descriptors against the original BRIEF [7] algorithm in addition to other well-known descriptor-based approaches, including SIFT [4], SURF [6], ORB [8], and FREAK [10] as implemented in the OpenCV library (version 2.4.3). We utilized the recommended default settings for each descriptor. SIFT and SURF were matched with the FLANN matcher, whereas a brute-force search was applied to BRIEF, ORB, and FREAK. For ORB, we detected 2000 keypoints (instead of the default 500), and we applied a pyramid scheme (with 8 levels) to achieve scale invariance. We rejected matches with distances larger than 3 $\times$ the best match. The accuracy and precision of the evaluations are shown in Fig. 2(a). It is evident that our BRIEF descriptors are able to achieve a similar performance to SIFT while significantly improving the accuracy and precision of the original BRIEF descriptor by over 40%.

Fig. 2(b) presents a break-down of the time (per frame in milliseconds) required to perform keypoint detection, feature extraction, feature matching and pose estimation for all the descriptors. Note that, for un-optimized implementations of our BRIEF descriptors, we are able to achieve a similar accuracy with regard to an optimized SIFT implementation at a tenth of the processing time.

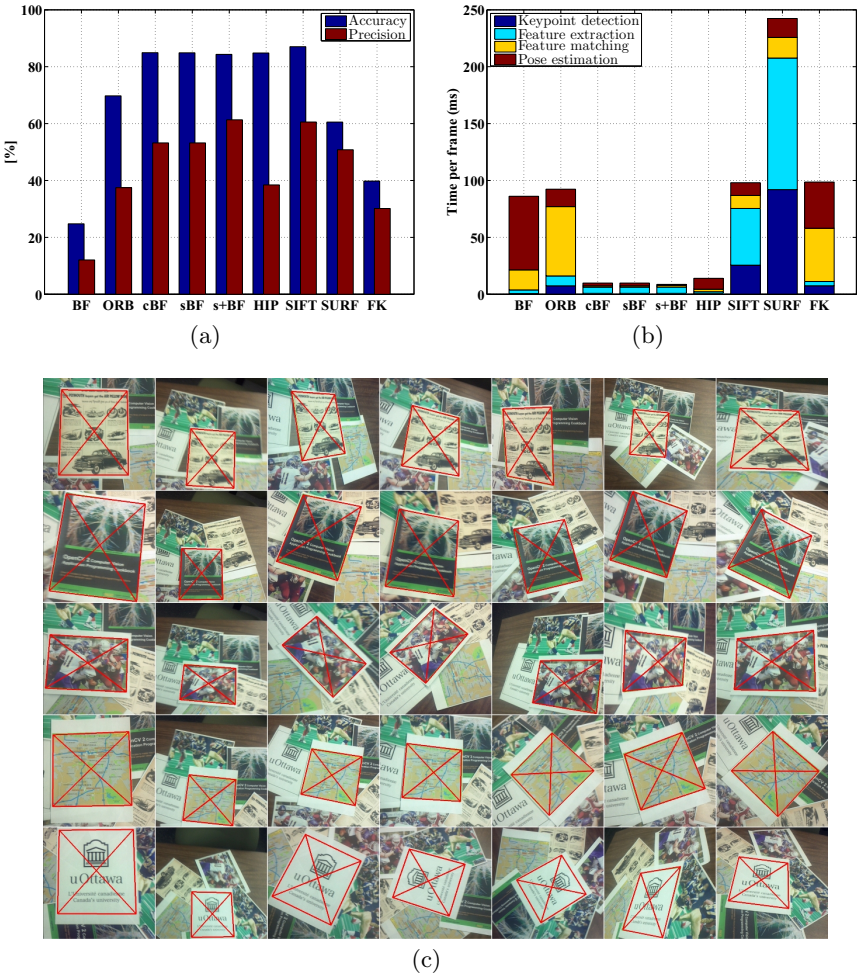

Fig. 2. Comparison results. (a) Accuracy and Precision, (b) Time per frame (milliseconds) for the different components during matching, (c) Sample detection results using cBRIEF. BF and FK stands for BRIEF and FREAK.

Interestingly, SIFT is more optimized than SURF in the current OpenCV implementation, hence it yields smaller computational times compared to SURF. Fig. 2(c) illustrates a sample of our detection results using cBRIEF.

9 Conclusion

We have developed a model generation framework to improve the robustness and efficiency of relatively weak descriptor-based approaches. Our model generation framework was successfully applied to the BRIEF descriptor resulting in significant performance gains comparable to SIFT. The viewset images are synthesized

by applying a perspective transformation, which enables us to sample the deformation space in a more controlled way requiring less images per viewpoint bin to cover a specified range of deformations. Hence, we do not require more than 3 minutes to generate a model as compared to the 20 minutes reported by Taylor et al. [14]. Currently our framework is limited to planar targets, however, in the future we would like extend our work to handle 3D targets as well.

References

1. 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)
2. Agrawal, M., Konolige, K., Blas, M.R.: Censure: Center Surround Extremas for Realtime Feature Detection and Matching. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 102–115. Springer, Heidelberg (2008)
3. Mair, E., Hager, G.D., Burschka, D., Suppa, M., Hirzinger, G.: Adaptive and generic corner detection based on the accelerated segment test. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part II. LNCS, vol. 6312, pp. 183–196. Springer, Heidelberg (2010)
4. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comp. Vision. 2, 91–110 (2004)
5. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Trans. Pattern Anal. Mach. Intell. 27(10), 1615–1630 (2005)
6. Bay, H., Ess, A., Tuytelaars, T., Gool, L.V.: Surf: speeded up robust features. Comp. Vis. and Img. Under. 110(3), 346–359 (2008)
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: IEEE International Conference on Computer Vision, pp. 2564–2571 (2011)
9. Leutenegger, S., Chli, M., Siegwart, R.Y.: BRISK: Binary Robust Invariant Scalable Keypoints. In: IEEE International Conference on Computer Vision, pp. 2548–2555 (2011)
10. Alahi, A., Ortiz, R., Vandergheynst, P.: FREAK: Fast Retina Keypoint. In: IEEE International Conference on Computer Vision and Pattern Recognition (2012)
11. Lepetit, V., Pilet, J., Fua, P.: Point matching as a classification problem for fast and robust object pose estimation. In: IEEE International Conference on Computer Vision and Pattern Recognition, pp. 244–250 (2004)
12. Lepetit, V., Fua, P.: Keypoint recognition using randomized trees. IEEE Trans. Pattern Anal. Mach. Intell. 28(9), 1465–1479 (2006)
13. Özuysal, M., Fua, P., Lepetit, V.: Fast keypoint recognition in ten lines of code. In: IEEE International Conference on Computer Vision and Pattern Recognition (2007)
14. Taylor, S., Rosten, E., Drummond, T.: Robust feature matching in 2.3μs. In: IEEE CVPR Workshop on Feature Detectors and Descriptors: The State Of The Art and Beyond (2009)
15. Taylor, S., Drummond, T.: Binary Histogrammed Intensity Patches for Efficient and Robust Matching. Int. J. Comp. Vis. 94(2), 241–265 (2011)
16. Gionis, A., Indyk, P., Motwani, R.: Similarity search in high dimensions via hashing. In: International Conference on Very Large Data Bases, pp. 518–529 (1999)