Interdisciplinary Information Sciences Vol. 18, No. 2 (2012) 107–112
©Graduate School of Information Sciences, Tohoku University
ISSN 1340-9050 print/1347-6157 online
DOI 10.4036/iis.2012.107

Learning to Identify Stable Keypoints

Yen-Cheng LIU and Hwann-Tzong CHEN*

Department of Computer Science, National Tsing Hua University,
No. 101, Section 2, Kuang-Fu Road, Hsinchu, 30013 Taiwan

Received May 14, 2012; final version accepted August 30, 2012

Through our experiments with the popular SIFT-DoG keypoint detector, we find that its stability in extracting keypoints from rotated images is good, but sometimes not as good as we expect. This paper presents our endeavor to improve the stability of the DoG keypoint detector by learning from tens of millions of training samples. The learning problem is formulated in a filtering setting, where the training samples are drawn from an oracle instead of using a fixed training set. We show that, by increasing the stability of keypoint detector, we may obtain discriminative local features for matching. The matching accuracy can be improved by 10% using the learned decision function as a watchdog to block unstable keypoints, with acceptable overheads in computation.

KEYWORDS: keypoint detection, boosting methods, sparse linear discriminant analysis

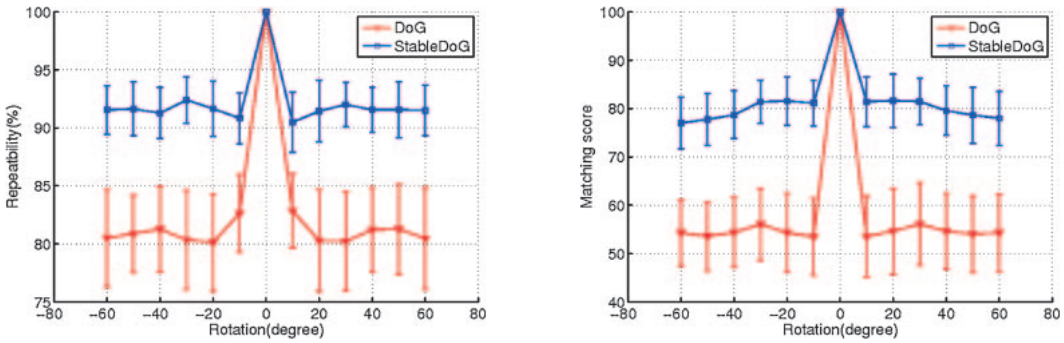
1. Introduction

Local features are often used as building blocks for solving higher-level tasks such as image matching, object detection, object recognition, and visual tracking. Lots of local feature detectors have been proposed over recent years, for example, saliency detector [7], MSER [9], SIFT-DoG [8], affine invariant detectors [10, 11], and SURF [1]. SIFT [8] is probably the most popular method to detect and describe local features. In the detection stage, SIFT uses the Difference-of-Gaussians (DoG) detector to identify keypoints in images. Although the DoG keypoint detector is able to find scale-invariant extrema in scale space, we find that it does not achieve satisfactory stability in extracting keypoints from rotated images. The problem can be illustrated by the curves shown in Fig. 1. The *repeatability* and the *matching score* are two measures presented in [12] for evaluating the performance of keypoint detectors in the task of matching two images. If most of the pairs of corresponding keypoints can be successfully detected in both images (the original image and the rotated one), the repeatability will be high. The repeatability only takes account of the locations and scales of the detected keypoints, regardless of the similarities between the image features of keypoints. The matching score, on the other hand, measures the distinctiveness of the detected keypoints. On computing the matching score, the local features around detected keypoints are represented by the SIFT feature descriptors, and the matching is performed by comparing the SIFT descriptors of the keypoints in the two images. If the detected keypoints do not produce distinctive descriptors, the correspondences between the keypoints in the two images would likely be wrong, due to incorrect matching between their SIFT descriptors. The evaluation in Fig. 1 is based on 30 randomly-chosen images, and for each image we rotate it by an angle from -60° to $+60^\circ$. We evaluate the DoG detector on each pair of the rotated image and the original image (rotation $= 0^\circ$). As can be seen in Fig. 1, the DoG detector does not perform very well on finding the correspondences between the original image and the rotated images. The average repeatability percentages are around 80% while the average matching scores are only around 55%.

However, if we remove those points that miss at least once in the rotations, the repeatability by definition will increase. Moreover, we observe that the matching score will also substantially increase. The results are illustrated by the blue curves in Figs. 1(a) and 1(b). Such a phenomenon is not simply due to the decrease in the number of detected keypoints. (Usually hundreds of keypoints remain after we remove the unstable ones.) The phenomenon might suggest that if a keypoint is stable under orientation changes, it would be more likely to contain distinctive features, and thus would help to improve the matching score. One of the explanations of this phenomenon is that SIFT needs to estimate the orientation of the detected keypoint before computing its feature descriptor, and an incorrect estimation of orientation would result in a totally different SIFT descriptor. We think that rotationally stable keypoints probably provide more useful information about the orientation.

Therefore, if we carry out a pre-screening step in which we rotate an image by different angles and eliminate those unstable (missing) keypoints, we may obtain a set of keypoints that will be more discriminative for later use in image matching or object recognition tasks. However, it is impractical and time-consuming to test the stability of the

* Corresponding author. E-mail: htchen@cs.nthu.edu.tw



(a) Repeatability under rotation (b) Matching score under rotation

Fig. 1. The evaluation is based on 30 randomly-chosen images. We rotate each image by an angle from -60° to $+60^\circ$, and then evaluate the DoG detector on each pair of the rotated image and the original image (0°). The DoG detector does not perform very well on finding the correspondences between the original image and the rotated images. The average repeatability percentages are around 80%, as illustrated by the red curves in (a), while the average matching scores are only around 55% [red curves in (b)]. By removing the points that are missing in at least one of the rotations, we observe that the repeatability percentages increase by definition [blue curves in (a)]. What is more, we find that the matching score will also substantially increase, as illustrated by the blue curves in (b). This phenomenon brings up a question: do rotationally-stable keypoints imply distinctive features?

keypoints through rotating an image by different angles. In this paper we present a learning-based method to address the problem, and we attempt to train a decision function for identifying stable keypoints using tens of millions of training samples.

The stability of keypoint detectors has been addressed in [3] and [5]. Dorkó and Schmid [3] propose to use the *description stability* as a criterion for scale selection. Their approach needs to compute the SIFT descriptors under different scales, and then measures the difference between SIFT descriptors to find the most stable scale. Such an approach cannot be used to determine the rotational stability of keypoints since the SIFT descriptors themselves might be extracted based on incorrect estimations of orientations, as discussed earlier. Gao and Vasconcelos present the *discriminant saliency detector* [4, 5] based on DCT coefficients and salient features. They introduce a discriminant formulation for interest-point detectors, and model the feature saliency as the discriminant power of a feature with respect to object classification. However, the analysis presented in [5] is only based on the repeatability; the performance regarding the matching score is not addressed. In this work we analyze the rotational stability of keypoint detectors in terms of the repeatability and the matching score.

The main contributions of our work are as follows. First, we address the rotational stability of keypoint detectors, and look into the correlation between stable keypoints and distinctive features. Second, we propose to learn a decision function for predicting the stability of keypoints. Such learning problems are not straightforward. To model the variations in local features and to prevent overfitting, we need to train the decision function using as many training samples as possible. It is impossible to manually label the training samples in this case. We formulate the learning problem in a filtering setting based on the FilterBoost algorithm [2], where the labeled training samples are drawn from an oracle instead of using a fixed training set. We devise an oracle that can continuously generate stable and unstable keypoints as positive and negative training samples. It is the combination of the FilterBoost algorithm with our oracle that makes the task of modeling rotationally-stable keypoints feasible.

2. Learning from an Oracle

The main idea of our approach is using FilterBoost [2] to learn a predictor for identifying rotationally-stable DoG keypoints. Unlike general Boosting algorithms that work with a fixed set of training data, FilterBoost is able to train the predictor over a very large or even infinite number of data provided by an oracle.

2.1 The design of the oracle

Given a training image, we rotate it by an angle of -30°, -20°, -10°, 0°, $+10^\circ$, $+20^\circ$, $+30^\circ$. We use the DoG detector to detect the keypoints in each of the rotated images. The keypoints that correspondingly appear in all rotated images under all rotation angles are considered stable and hence are labeled as positive data. The rest keypoints are considered unstable and thus labeled as negative data. We extract the patches $\{\mathbf{x}\}$ around the keypoints from the reference image (0°) according to the scales estimated by the SIFT-DoG algorithm. Each patch $\mathbf{x}$ is associated with a label y, which indicates that the corresponding keypoint is stable ($y = 1$) or unstable ($y = 0$). Given an abundant supply of natural images, the oracle can easily produce tens of millions of stable and unstable keypoint patches. The patches produced by the oracle are passed to FilterBoost for training the predictor.

2.2 FilterBoost—Filtering phase

Unlike AdaBoost, which always trains the predictor with a fixed set of samples, FilterBoost continuously accepts important samples from oracle, and uses those samples to simulate the real distribution of data. FilterBoost estimates the probability of samples by

$$q_t(\mathbf{x}, y) = \frac{1}{1 + e^{yF_t(\mathbf{x})}}, \tag{1}$$

where $\mathbf{x}$ is a training sample produced by the oracle, y is the label of $\mathbf{x}$, and $F_t(\mathbf{x})$ is the strong classifier at the current iteration t given by

$$F_t(\mathbf{x}) = \sum_{k=1}^{t-1} \alpha_k h_k(\mathbf{x}). \tag{2}$$

The strong classifier is a linear combination of weak learners $h_k(\mathbf{x})$ with weights α_k available thus far. The mechanism of FilterBoost is to draw new training samples according to $q_t(\mathbf{x}, y)$. Samples that are hard to classify would have a higher probability to be accepted. Through the filtering scheme we obtain a set of samples, and we may use these samples to find a weak learner $h_t(\mathbf{x})$ at iteration t. We may apply the same scheme, and acquire a new set of samples to compute the weight α_t for $h_t(\mathbf{x})$.

2.3 FilterBoost—Boosting phase

The samples acquired from the previous filtering phase with the oracle are used to find a new weak learner $h_t(\mathbf{x})$. We employ *sparse linear discriminant analysis* (SLDA) [13] to construct weak learners. Given a set of training samples, we solve the corresponding SLDA problem to obtain a 1D projection that maximizes the generalized Rayleigh quotient subject to a sparsity constraint. We then project all training samples onto the 1D subspace, and decide a threshold (using the sorting tactic in [14]) to separate the training samples into two classes. We apply the greedy *forward selection* algorithm suggested in [13] to solve SLDA. Moreover, since the weak learners are binary classifiers, we can further apply the efficient eigenvalue computation based on rank-one update and the partitioned matrix inverse technique [13]. The advantage of using SLDA to construct weak learners is that the evaluation of a weak learner would involve only a few pixels owing to the sparsity constraint. The evaluation is hence very efficient.

Let $X = \{\mathbf{x}\}$ be the current set of training patches and Y the set of the corresponding binary labels. Similar to general booting algorithms, FilterBoost aims to find a weak learner $h_t(\mathbf{x})$ that minimizes the current classification error $\Pr_{\{X,Y\}}[h_t(\mathbf{x}) \neq y]$ over the current training samples $\{X, Y\}$. The new strong classifier is expressed as

$$F_{t+1}(\mathbf{x}) = F_t(\mathbf{x}) + \alpha_t h_t(\mathbf{x}), \tag{3}$$

where the new weight α_t is also computed via a filtering scheme. The algorithm for computing α_t and the theoretical details of FilterBoost can be found in [2].

The final decision function learned by FilterBoost is

$$H(\mathbf{x}) = \operatorname{sign}\left(\sum_{k=1}^{T} \alpha_k h_k(\mathbf{x})\right), \tag{4}$$

where T is the total number of weak learners. We may predict the stability of an input keypoint patch using the learned decision function $H(\mathbf{x})$.

3. Experimental Results

We use the oracle described in §2 to identify stable and unstable keypoints from the images of Caltech-256 Object Category Dataset [6]. The corresponding image patches are extracted from the locations of keypoints to produce training samples. In our experiment, the oracle totally produces 14,381,979 samples, and after filtering, the number of samples that are accepted for training is 8,727,397. Note that it is impractical to train a predictor directly on such a large training set using SVM or AdaBoost. As described in the previous section, FilterBoost incorporates a filtering scheme to accept samples from oracle with probability, which is estimated by the weak learners available so far. At each iteration FilterBoost works on a feasible size of training set, and the overall distribution of data is well modeled by the filtering scheme. Specifically we let FilterBoost accept $10000 \cdot \log(t + 1)$ samples at iteration t. Each training patch is downsampled to 49-by-49 pixels and represented as a 2401-dimensional vector for computing SLDA. To enforce the sparsity, we require that only 300 out of the 2401 pixel positions may have nonzero coefficients. The number of weak learners is $T = 200$. It takes one day to learn a decision function via FilterBoost on a 3 GHz PC.

The learned decision function can then be used to predict the stability of a keypoint detected by the DoG detector. We perform our test in Matlab. Typically, detecting 3000 keypoints from an image by DoG takes about 2 seconds, and predicting the stability of those keypoints by the learned decision function needs about 7 seconds. For evaluation, we choose eight categories of Microsoft Research Cambridge Object Recognition Image Database to generate our test

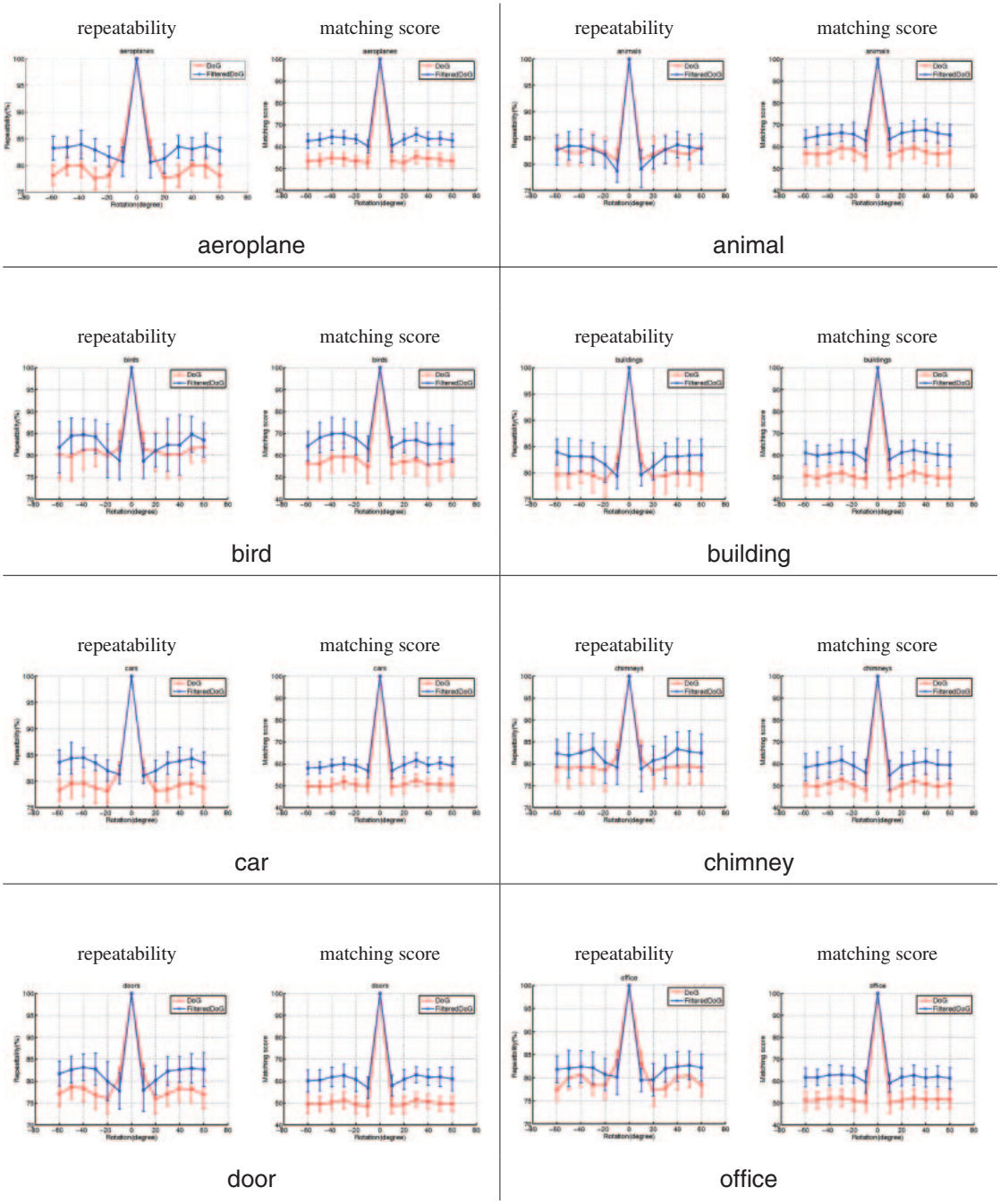


Fig. 2. Evaluations of keypoint detection under orientation change using eight-category images from the Microsoft Research Cambridge Object Recognition Image Database. From each category we select 30 images, and rotate each image by an angle of $-60°, -50°, \ldots, +60°$. For each rotated image, the repeatability and the matching score of the keypoints are computed with respect to the original image (rotation $= 0°$). The red curves are the results of DoG keypoints. The blue curves are the results of filtered stable DoG keypoints using our method. In most cases, our method is able to increase the matching score by about 10%.

Table 1. The average repeatability and matching score over all rotation angles.

	aeroplane mean% (std)	animal	bird	building	car	chimney	door	office
DoG_{rep}	79.45 (2.72)	82.25 (2.67)	80.79 (4.29)	79.92 (2.79)	79.40 (2.65)	79.42 (4.03)	77.97 (3.34)	79.61 (3.19)
Ours_{rep}	82.58 (2.61)	82.19 (3.11)	82.32 (5.04)	82.34 (3.16)	83.15 (2.32)	81.63 (4.47)	81.37 (4.02)	81.47 (3.43)
DoG_{mat}	53.81 (2.89)	57.44 (5.20)	57.07 (6.98)	50.43 (4.11)	50.45 (3.08)	50.27 (4.89)	49.83 (3.75)	51.36 (4.24)
Ours_{mat}	63.19 (3.24)	65.48 (4.73)	66.30 (7.11)	60.43 (5.09)	59.07 (3.35)	59.11 (6.38)	60.75 (4.94)	61.63 (4.44)

Table 2. The average ratio of "filtered stable keypoints" to "DoG keypoints" over all rotation angles.

	aeroplane	animal	bird	building	car	chimney	door	office
Ratio	0.69	0.58	0.56	0.64	0.68	0.66	0.60	0.67

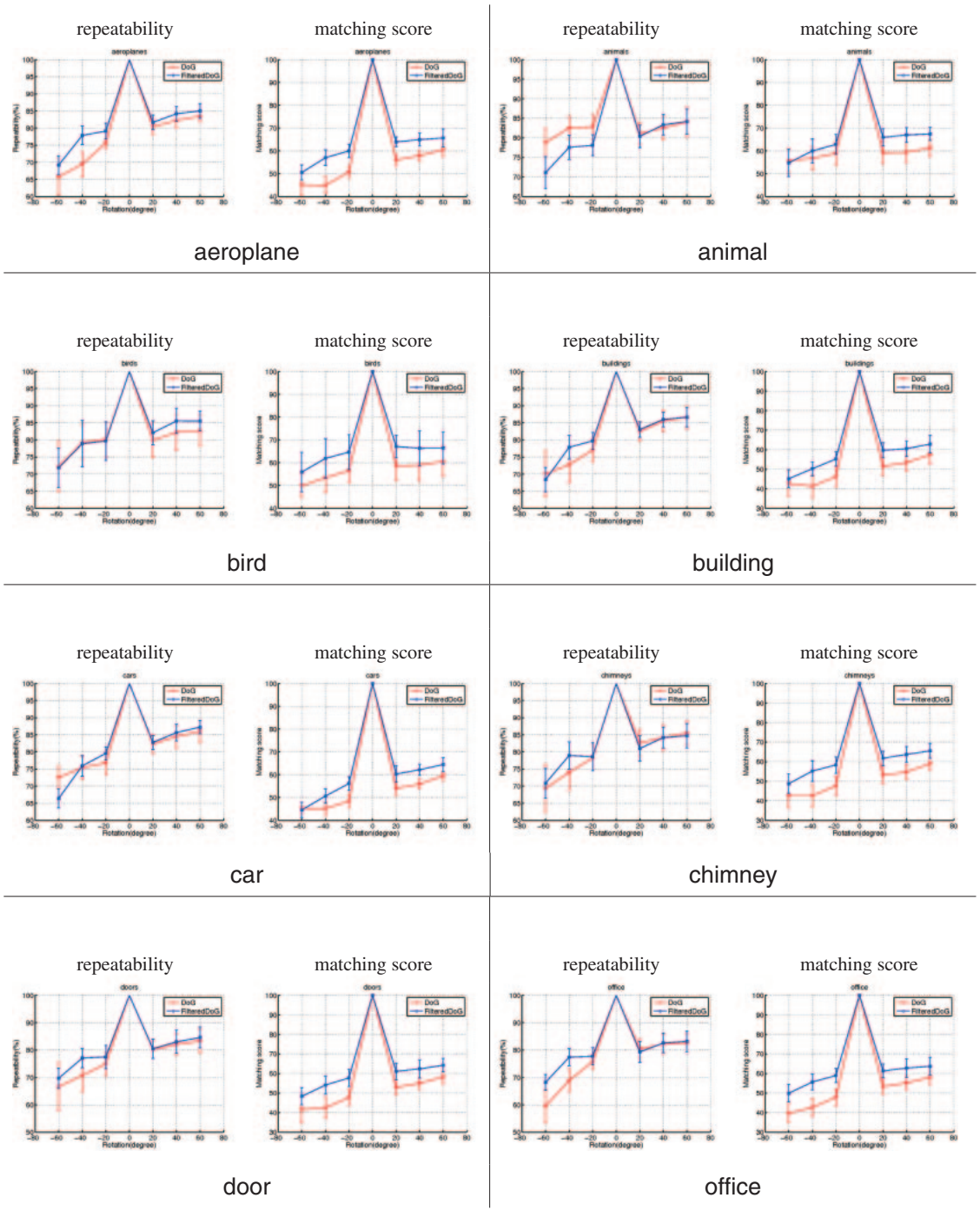


Fig. 3. Evaluations of keypoint detection under orientation and scale changes. From each of the eight categories of MSRC images, we select 30 images. Each image is rotated by an angle of $-60°$, $-40°$, $-20°$, $+20°$, $+40°$, $+60°$ and re-scaled by a factor of 0.75, 0.85, 0.9, 1.1, 1.15, 1.25. The repeatability and the matching score of the keypoints are computed with respect to the original image (rotation $= 0°$, scaling $= 1$). The red curves are the results of DoG keypoints. The blue curves are the results of filtered stable DoG keypoints using our method. The improvement in repeatability is marginal in comparison with the previous rotation-only case, but the improvement in matching score is still significant.

images. The eight categories are aeroplane, animal, bird, building, car, chimney, door, and office. From each category, we select 30 images, and rotate each image by an angle of $-60°, -50°, \ldots, +60°$. We use the DoG detector to extract keypoints from each test image, and then use the learned decision function to remove unstable keypoints.

As mentioned in §1, we employ the *repeatability* and the *matching score* [12] as the quality measures for evaluating the performance of keypoint detection under different rotations and scales. The repeatability score for a given pair of images is computed as the ratio between the number of *keypoint correspondences* and the smaller of the number of detected keypoints in the pair of images. The matching score is computed as the ratio between the number of *correct matches* and the smaller number of detected keypoints in the pair of images. A match is defined as the nearest neighbor in the descriptor space. In particular, a high matching score implies that the detected keypoints are discriminative and more useful for subsequent matching tasks. The details of computing the repeatability and the matching score can be found in [12], and the related evaluation software has been made available online by the authors of [12].

For each rotated image, the repeatability and the matching score of the keypoints are computed with respect to the original image (rotation $= 0^\circ$). The performances of the DoG detector and the learned decision function are shown in Fig. 2. The red curves are the results of DoG keypoints, and the blue curves are the results of filtered stable keypoints using our method. Table 1 summarizes the average repeatability and matching score over all rotation angles. In most cases, our method is able to increase the matching score by about 10%, which means that the filtered stable keypoints identified by the learned decision function are indeed more distinctive. The ratios of "filtered stable keypoints" to "DoG keypoints" are listed in Table 2. Our method generally keeps more than half of the number of original DoG keypoints. We further test the learned decision function under the combined transformations of rotation and scaling. We generate rotated and re-scaled images from the eight categories of the MSRC image database. Each image is rotated by an angle of -60°, -40°, -20°, $+20^\circ$, $+40^\circ$, $+60^\circ$ and re-scaled by a factor of 0.75, 0.85, 0.9, 1.1, 1.15, 1.25 correspondingly. The performances on these images are shown in Fig. 3. The improvement in repeatability is marginal in comparison with the previous rotation-only case, but the improvement in matching score is still significant.

4. Conclusion

We have shown that we are able to improve the accuracy of feature matching by excluding unstable keypoints. The experimental results suggest that stable keypoints may correlate with distinctive features, and therefore are more useful for image matching or other high-level tasks. In this regard we present a learning-based method to identify stable and unstable keypoints. The underlying learning problem requires as many training samples as possible to model the wide variations in local features and to prevent overfitting. We employ the FilterBoost algorithm and devise an oracle that can infinitely generate stable and unstable keypoints as positive and negative training samples. The combination of the FilterBoost algorithm and our oracle makes the task of learning stable keypoints feasible. The learned decision function can effectively predict the stability of keypoints, and our method may improve the matching score of keypoints in most cases.

REFERENCES

[1] Bay, H., Tuytelaars, T., and Gool, L. J. V., "Surf: Speeded up robust features," in *ECCV (1)*, 404–417 (2006).
[2] Bradley, J. K., and Schapire, R., "Filterboost: Regression and classification on large datasets," in *Advances in Neural Information Processing Systems 20*, pp. 185–192, Cambridge, MA (2008).
[3] Dorkó, G., and Schmid, C., "Maximally stable local description for scale selection," in *ECCV (4)*, 504–516 (2006).
[4] Gao, D., and Vasconcelos, N., "Discriminant saliency for visual recognition from cluttered scenes," in *NIPS* (2004).
[5] Gao, D., and Vasconcelos, N., "Discriminant interest points are stable," in *CVPR* (2007).
[6] Griffin, G., Holub, A., and Perona, P., Caltech-256 object category dataset, Technical Report 7694, California Institute of Technology (2007).
[7] Kadir, T., and Brady, M., "Saliency, scale and image description," *International Journal of Computer Vision*, **45(2)**: 83–105 (2001).
[8] Lowe, D. G., "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, **60(2)**: 91–110 (2004).
[9] Matas, J., Chum, O., Urban, M., and Pajdla, T., "Robust wide baseline stereo from maximally stable extremal regions," in *BMVC* (2002).
[10] Mikolajczyk, K., and Schmid, C., "Scale & affine invariant interest point detectors," *International Journal of Computer Vision*, **60(1)**: 63–86 (2004).
[11] Mikolajczyk, K., and Schmid, C., "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, **27(10)**: 1615–1630 (2005).
[12] Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., and Gool, L. J. V., "A comparison of affine region detectors," *International Journal of Computer Vision*, **65(1–2)**: 43–72 (2005).
[13] Moghaddam, B., Weiss, Y., and Avidan, S., "Fast pixel/part selection with sparse eigenvectors," in *ICCV*, 1–8 (2007).
[14] Viola, P. A., and Jones, M. J., "Robust real-time face detection," *International Journal of Computer Vision*, **57(2)**: 137–154 (2004).