Large scale evaluation of local image feature detectors on homography datasets

Karel Lenc
http://www.robots.ox.ac.uk/~karel/
Andrea Vedaldi
http://www.robots.ox.ac.uk/~vedaldi/

Department of Engineering Science
University of Oxford
Oxford, UK

Abstract

We present a large scale benchmark for the evaluation of local feature detectors. Our key innovation is the introduction of a new evaluation protocol which extends and improves the standard detection repeatability measure. The new protocol is better for assessment on a large number of images and reduces the dependency of the results on unwanted distractors such as the number of detected features and the feature magnification factor. Additionally, our protocol provides a comprehensive assessment of the expected performance of detectors under several practical scenarios. Using images from the recently-introduced *HPatches* dataset, we evaluate a range of state-of-the-art local feature detectors on two main tasks: viewpoint and illumination invariant detection. Contrary to previous detector evaluations, our study contains an order of magnitude more image sequences, resulting in a quantitative evaluation significantly more robust to over-fitting. We also show that traditional detectors are still very competitive when compared to recent deep-learning alternatives.

1 Introduction

Despite advances in distributed representations such as deep convolutional networks, local viewpoint invariant features still play an important role in tasks such as structure from motion and image retrieval. In these applications, deep learning has often been used to *improve* rather than to *replace* local features. While most of this work focused on learning feature *descriptors*, recently there has been progress in learning *detectors* as well. For example, in [36] use deep networks to learn a local feature detector robust to illumination changes, [38] for orientation assignment, [15] for learning detectors without supervision, and [37] for learning local feature detectors, orientation assignment and descriptors.

An obstacle to further progress in learning local feature detectors is the lack of a modern, large-scale evaluation benchmark for this task. Advances in tasks such as image classification were driven by the introduction of benchmarks such as ImageNet. For local feature descriptors, recent contributions such as HPATCHES [3] may play a similar role, but there is still no good solution for detection. Several works for testing performance of both detector and descriptors emerged, [23, 31], however we believe that being able to test and compare algorithms separately provides invaluable insight into where the progress is made.

In order to address this shortcoming, in this paper we propose **a modern evaluation of feature detectors**. We do so by augmenting the evaluation protocol (section 3) of feature benchmarks which come with ground truth homographies for image sequences representative of various difficult imaging scenarios, such as illumination and viewpoint changes. We build especially on the recently-introduced HPATCHES dataset; however, while the latter contains pre-detected image patches for descriptor evaluation, we discard such patches and use the images as a whole to assess feature detectors instead. We further refer to this dataset as HPSequences.

For the evaluation protocol, we start from the detector repeatability evaluation protocol introduced in the classic paper of [22], as it is an accepted standard, and we improve it in various ways. Specifically, compared to earlier benchmarks such as VGG Affine, which are nowadays heavily over-fitted due to their small size and due to having been used by the community for many years, HPSequences is much larger and less prone to over-fitting. Furthermore, we improve the evaluation protocol by addressing issues in the invariance to feature magnification factor found in the reference implementation of repeatability (section 3.1). We also propose to modify the protocol to explicitly control for the number of detected features per image (section 3.2), yielding fairer detector comparisons. Due to the significantly increased number of images compared to VGG Affine (696 vs 48), we also change the way results are aggregated, reported, and analysed, comparing detectors quantitatively using a single plot (section 3.3). Additionally, we include trivial baselines based on random features which provide lower bounds of the expected performance. The new benchmarking code for automatic evaluation of detectors is released in the open source domain, simplifying reproducibility of the future research. We aim to provide a robust, easy-to-reproduce and easy-to-use evaluation platform for comparison of local feature detector performance on planar scenes. Both source code and pre-computed scores used for this manuscript are freely available[1].

Having designed a suitable benchmark, our second contribution is to **analyse** classic feature detector against modern ones based on **deep learning** (section 5). We find that learning detectors significantly improves robustness to illumination changes, but that, for viewpoint invariance, traditional detectors using scale selection and affine adaptation are still nearly as good and sometimes better than learned ones.

2 Related work

In this section we introduce evaluated local feature detectors (section 2.1) and existing benchmarks for their evaluation (section 2.2).

2.1 Local detectors

Local image feature detectors differ by the type of features that they extract, *e.g.* points [13, 28], circles [17, 19], or ellipses [5, 18, 20]. In turn, the type of feature determines which class of transformations that they can handle: Euclidean transformations, similarities, and affinities respectively. Additionally, we can divide detectors as follows:

Hand-crafted detectors. Standard, hand-crafted local feature detectors vary based on the visual structures used as anchors for the features, *e.g.* corners or other operators of the image

[1] https://github.com/lenck/vlb-deteval

intensity such as the *Hessian of Gaussian* [7] or the *structure tensor* [12, 13, 42]. Going beyond roto/translation, scale selection methods using the *Laplacian/Difference of Gaussian* operator (L/DoG) or Hessian of Gaussian were introduced in [17, 19] and further extended with *affine adaptation* [5, 20] to handle full affine transformations.

Accelerated detectors. Machine learning can be used to imitate and accelerate an off-the-shelf detector defined *a-priori* [10, 14, 29, 32]. Rosten et al. [28] use simulated annealing to optimise the parameters of their FAST detector for repeatability. For the SURF detector [6], the authors use integral images to approximate the Hessian feature response.

Learned detectors. Learning detectors attempts to discover or improve the visual anchors used for detection, a task much harder than using hand-crafted anchors. Early attempts used genetic programming [25, 33]. More recently, Yi et al. [38] learn to estimate the orientation of feature points using deep learning. A related approach is the TILDE detector [36] for illumination invariance. The LIFT framework [37] aims at learning detector, descriptor and orientation estimation jointly using patches, while SuperPoint [9] uses full images. Another approach to unsupervised learning of keypoint detectors is DNET [15], which is trained using the covariance constraint and no supervision. A version of this detector is TCDET [39], combined geometry and appearance losses. The covariant constraint is extended for affine adaptation in [24].

2.2 Evaluation of local detectors

The standard protocols for the evaluation of local feature detectors and descriptors was established by [21, 22] using the VGG Affine dataset, which contains 8 sequences of 6 images related by a known homography. Detectors are assessed in terms of their *repeatability*, which measures their robustness to nuisance effects such as a change in viewpoint or illumination. The standard definition of repeatability has some shortcomings. First, features are compared by the overlap of their support, generally elliptical, which may not encode all relevant geometric information (e.g. it disregards the feature orientation) and depends on the size of regions, which is arbitrary and requires normalisation. Second, computing repeatability is somewhat slow and uses in practice a number of approximations, which we show in this paper are not innocuous.

Many datasets followed the introduction of VGG Affine. In the *Hanover dataset* [8], the number of sequences is extended while improving the precision of the homography. While the traditional and most commonly used VGG Affine dataset contains images that are all captured by a camera, the *Generated Matching dataset* [11] is obtained by generating images using synthetic transformations. The *Edge Foci dataset* [40] consists of sequences with very strong changes in viewing conditions, making the evaluation somewhat specialized to extreme cases; furthermore, the ground truth for non-planar scenes does not uniquely identify the correspondences since the transformations cannot be well approximated by homographies. In the *Webcam dataset* [36], new sequences for testing illumination changes are presented. The *DTU robots dataset* [1] goes beyond homographies and uses scenes with a known 3D model, obtained using structured lighting. In [23], the authors introduce a new dataset for generalised wide baseline stereo matching across geometry (homography and epipolar), illumination, appearance over time and capturing modes.

Instead of introducing new evaluation protocols, our main goal is to provide a large scale evaluation baselines over multiple datasets. Additionally we improve the repeatability score and quantitatively analyse results across different tasks.

3 Evaluation protocol: repeatability revisited

In this section we refresh the traditional repeatability evaluation method proposed by Mikolajczyk et al. [22], addressing some of its shortcomings and improving its applicability to large datasets.

Given an image I, a detector extracts a set $\mathcal{D} = \{\mathcal{R}_1, \dots, \mathcal{R}_n\}$ of n regions $\mathcal{R}_i \subset \mathbb{R}^2$, generally ellipses. Given a second image I' related to the first by an homography, the same detector extracts another list $\mathcal{D}' = \{\mathcal{R}'_1, \dots, \mathcal{R}'_n\}$ of m regions. Following Mikolajczyk et al. [22], the repeatability score $rep(\mathcal{D}, \mathcal{D}', H)$ for the detected features is the fraction of features that match between images with sufficient geometric overlap up to the homography H. While the concept is simple, there are many important implementation details that strongly affect the outcome.[2] These details are discussed next.

The degree of geometric match between two regions $\mathcal{R}, \mathcal{Q} \subset \mathbb{R}^2$ is given by their *overlap* $o(\mathcal{R}, \mathcal{Q}) = |\mathcal{R} \cap \mathcal{Q}| / |\mathcal{R} \cup \mathcal{Q}|$. If H is the homography transformation that reprojects pixels from image I' back to image I, the overlap measure can be changed to $o(\mathcal{R}, H\mathcal{R}')$ to compensate for this transformation. However, as noted by Mikolajczyk et al. [22], overlap can generally be increased just by scaling (magnifying) the detected features by a constant *magnification factor* $s \in \mathbb{R}_+$, which can be trivially incorporated in the definition of any detector. For example, if $s\mathcal{R}$ denotes the effect of scaling the region $\mathcal{R}$ by a factor s around its center of mass, and if $\mathcal{R}$ and $\mathcal{R}'$ differ only by a shift, then $\lim_{s\to\infty} o(s\mathcal{R}, Hs\mathcal{R}') = 1$. Mikolajczyk et al. [22] address this issue by rescaling features so that the first one has an area of 30^2, resulting in the (asymmetric) normalised overlap score $o(\mathcal{R}, \mathcal{R}'|H) = o(s(\mathcal{R})\mathcal{R}, Hs(\mathcal{R})\mathcal{R}')$, where $s(\mathcal{R}) = 30^2/|\mathcal{R}|$. Please note that this does not fully remove the influence of the detected scale, as the relative scale between the compared regions is still important (as the normalisation constant for both regions is $s(\mathcal{R})$).

Next, in order to compute repeatability in two feature sets, features must be matched based on ellipse overlap [3]. In order to do so, features that do not belong to the common part of I and I' are dropped as they cannot be matched. This is done by sending the center of each region $\mathcal{R}'$ to I using H and testing for inclusion in the domain of I; the same operation is repeated for regions $\mathcal{R}$ in the other direction. Let $\mathcal{D}_c$ and $\mathcal{D}'_c$ be the remaining features. Pairs of such regions are associated with score $s(\mathcal{R}_i, \mathcal{R}'_j) = o(\mathcal{R}_i, \mathcal{R}'_j|H)$ if their normalised overlap is at least $1 - \varepsilon_O$ and $s(\mathcal{R}_i, \mathcal{R}'_j) = -\infty$ otherwise. The matches $\mathcal{M}^* \subset \mathcal{D}_c \times \mathcal{D}'_c$ are determined as the bipartite graph that maximises[4] the overall score $\sum_{(\mathcal{R},\mathcal{R}')\in\mathcal{M}} s(\mathcal{R}, \mathcal{R}')$. Note that this maximization retains only pairs with overlap above the threshold and matches each region at most once. Finally, *repeatability* is defined as $rep(\mathcal{D}, \mathcal{D}', H) = |\mathcal{M}^*| / \min\{|\mathcal{D}_c|, |\mathcal{D}'_c|\}$.

3.1 Magnification factor invariance

While in principle the use of a normalised overlap measure should make repeatability invariant to the detector magnification factor, the reference implementation of this measure still has a strong empirical dependency on this parameter, as can be seen in fig. 1-left. We have identified that the cause of this issue is in the heuristic used for accelerating the ellipse overlap computation. This heuristic filters out ellipse pairs whose enclosing circles cannot

[2] The actual implementation slightly differs from the definition in the paper Mikolajczyk et al. [22] which also lacks some details; our description follows the authoritative implementation by the same authors.

[3] Several works [26, 28] use only distance of the keypoint centres, however it is only applicable for joint detector and descriptor evaluation.

[4] In practice, bipartite matching is approximated greedily.

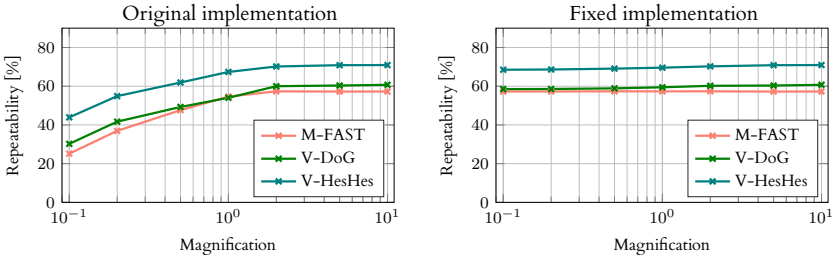

Figure 1: Average detector repeatability on VGG Affine for three detectors and increasing magnification factor (log scale). Due to normalisation, the lines should be approximately constant (right) but this is not the case in the original implementation (left) due to approximations.

overlap, a test which can be calculated quickly. However, in the original implementation of the test, this heuristic was applied *before* the ellipse normalisation step. This leads to ellipses with area smaller than 30^2 pixels being mistakenly skipped as unable to overlap, reducing the repeatability score. After fixing this relatively minor issue (by normalising the excircles), in fig. 1-right we show that the repeatability becomes invariant to the magnification factor.

3.2 Detection threshold

Many local feature detectors have a single parameter that controls selectivity, which we generically call *detection threshold* τ. In hand-crafted detectors, τ is usually the minimum value of the cornerness measure, such as DoG, Hessian or structure tensor *etc.*, for which a feature is retained.

One would expect a detector to provide stable performance across all its detection thresholds. In practice, however, this might not be reflected by repeatability. In fact, with an increased number of features, it becomes easier to match features by accident, making repeatability biased for settings that produce more features. That is why, for a fair comparison, local feature detectors need to return a similar number of features and we test random detectors to obtain a baseline performance.

Since each detection algorithm anchors features to different visual primitives in an image, the number of detected features cannot be equalized by choosing a constant τ per detector for the whole dataset. Instead, similarly to [23, 36], we run each detector to extract as many features as possible (by lowering τ) and then consider only the top-n detections from each image ranked by detection score, where $n \in \mathcal{N} = \{100, 200, 500, 1000\}$.[5] Testing different values of n is useful because the number of detections per image may differ based on the application and shows whether the detection score is predictive of the detected regions repeatability. As far as we are aware, testing the detector performance over various operational point is not a standard practice in local feature evaluation.

3.3 Aggregated metrics and their analysis

So far, we have explained how to compute repeatability for a pair of images. Here we look at how a large dataset of image pairs can be used for assessment.

[5]The upper limit 1000 was selected empirically, as some detectors produce fewer features even at the lowest τ than others.

Dataset	# Seq.	#Ims.	#Im. pairs
VGG Affine [22]	8	48	40
Webcam [36]	6	250	125
HPSequences [4]	116	696	580

Table 1: Basic statistics of the selected datasets for local feature detector evaluation.

The benchmark of Mikolajczyk et al. [22] contains a number of *image sequences* $(I_t)_{t=0}^{T}$ and their evaluation reports repeatability for each sequence, fixing I_0 as reference image and varying $I_t, t = 1,\dots,T$. Each sequence tests a particular aspect of feature detection, such as invariance to viewpoint, illumination, or noise changes. Furthermore, images in each sequence are sorted by the size of the nuisance variation, so plotting repeatability against t normally shows a progressive reduction in repeatability.

Such an approach is suitable for VGG Affine, which contains just 8 sequences with 6 images each. Clearly, however, it does not scale well to larger datasets. Furthermore, it does not provide a single performance metric per detector, nor corresponding confidence margins, which makes it difficult to compare detectors' performance and to know how significant the differences are. Another issue is that in datasets such as Webcam [36] and HPSequences [4] images cannot be easily sorted by the size of the nuisance variation, thus plotting repeatability against t is meaningless.

We approach this issue by computing aggregated statistics over multiple images and factors of variation, similar to [36, 38, 41]. As repeatability is very sensitive to the number of features extracted, we also compute an average over different detection thresholds and analyse the distributions of repeatability scores so obtained to extract confidence margins.

In more detail, we are given a dataset which consists of a set of image pairs and homographies $\mathcal{T} = \{(I_1, J_1|H_1), \dots, (I_T, J_T|H_T)\}$ (table 1). We will denote $rep(d,t,n)$ as the repeatability of a detector d, task $t \in \mathcal{T}$ and number of detections per image $n \in \mathcal{N}$. To score a detector s, we average repeatability across tasks and number of detections:

$$rep(d,n) = |\mathcal{T}|^{-1} \sum_{t\in\mathcal{T}} rep(d,t,n), \quad rep(d) = |\mathcal{N}|^{-1} \sum_{n\in\mathcal{N}} rep(d,n). \tag{1}$$

An ideal detector in our evaluation has a high average repeatability. Additionally, we consider also the *variance* of the repeatability score, as a low variance means that the detector performance is consistent across different cases. We visualise both average and variance using box-and-whisker diagrams, plotting repeatability on the x axis and detectors on the y axis (fig. 2). These diagrams summarise at a glance the statistics $rep(d,t,n)$ for each detector d. The box percentiles are 25% and 75% (first and third quartile) and the whisker percentiles are 10% and 90%. The length of the whiskers correspond to the length of the distribution tail. Additionally, we show the median (solid line) and the mean (red cross) of each distribution. We vary the line style of the whiskers to group detectors by type, generally based on their purported invariance (dotted for translation, dash-dot for scale, and dashed for affine invariant detectors). Finally, for each detector, we show $rep(d,n)$ using box markers: .1k for $rep(d,100)$, .5k for $rep(d,500)$, and 1k for $rep(d,1000)$.

Stability error across detection thresholds. To quantify the stability of the detector performance across detection thresholds, we calculate the detector instability as the standard deviation of the detector repeatability across different numbers of features, normalised by the average repeatability:

$$stb(d) = rep(d)^{-1} \cdot \sqrt{|\mathcal{M}|^{-1} \textstyle\sum_n [rep(d,n) - rep(d)]^2}. \tag{2}$$

Table 2: Tested local feature detectors and their speed (in seconds) on four test images from HPSequences (CPU: single thread Intel Xeon E5-2650 v4; GPU: NVIDIA Tesla M40).

Sequence # Pixels		AJUNTAMENT 0.3M		MELON 0.5M		WAR 1.0M		CONSTRUCTION 3.1M	
Detector	Impl.	CPU	GPU	CPU	GPU	CPU	GPU	CPU	GPU
FAST-T [28]	MATLAB	0.10	-	0.01	-	0.01	-	0.03	-
SURF-S [6]	MATLAB	0.14	-	0.12	-	0.42	-	1.20	-
BRISK-S [16]	MATLAB	0.26	-	0.26	-	0.37	-	0.52	-
DoG-S [17]	VLFeat [35]	0.14	-	0.17	-	0.53	-	2.80	-
Hes-A [20]	VLFeat [35]	0.55	-	0.56	-	2.67	-	11.91	-
TILDE-T [36]	[36]	3.42	-	5.42	-	9.38	-	34.99	-
LIFT-S [37]	[37]	-	149.94	-	155.41	-	163.28	-	223.52
DNET-T [15]	[15]	7.64	0.13	12.22	0.18	25.01	0.35	83.92	1.24
DNET-S [15]	[15]	15.24	0.35	24.06	0.49	49.64	0.90	465.42	3.00
TCDET-S [39]	[39]	3.67	1.42	7.04	1.67	13.24	2.25	40.47	5.35

4 Selected local feature detectors

Reference detectors. Due to large number of existing detectors, we select a sample representative of the breadth of possible approaches. Furthermore, we restrict our attention to detectors that associate a detection strength τ to each feature (possibly after modifying the implementation of the detector to expose such a value), as needed for selection in the evaluation protocol. That is why we exclude MSER [18][6] and Edge Based Regions [34].

The selected detectors are listed in Table 2. Detectors are suffixed with -T, -S, -A to emphasise their theoretical viewpoint invariance class (translation, translation+scale and affine respectively). We test a number of detectors representative of traditional techniques such as Harris/Laplace/Hessian cornerness/scale selection and affine adaptation (DoG-S — aka SIFT-S, SURF-S, Hes-A). We also test FAST-T and BRISK-S, which uses learning to accelerate a standard corner detector. Finally, we test several last-generation detectors that use deep learning: TILDE-T, TCDET-S, LIFT-S, DNET-T, DNET-S. DNET-S is a version of DNET [15] which is evaluated on scaled images, similarly as TCDET-S [39]. The table also reports their evaluation speed, as this is often a key parameter in applications. Unfortunately, for more recent works [9, 26, 30], the source code was not available at the time of publication.

Random baseline detectors. Detectors are also contrasted against a baseline obtained by sampling n features at random [36]. We consider: random points (RAND-T), circles (RAND-S) and ellipses (RAND-A). Given a scale s and a $H \times W$ image, the feature center (u,v) is obtained by sampling uniformly at random the set $[s, W-s] \times [s, H-s]$. The scale is sampled as $s \sim \min\{\|\mathcal{N}(s_{min}, (s_{max}-s_{min})^2/4)\|, s_{max}\}$ where $s_{min} = 0.1$ and $s_{max} = 50$ are the minimum and maximum scales. The normal distribution captures the fact that, for most detectors, less features are detected at larger scales. Finally, ellipses are generated by sampling the affine transformation $A = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \cdot \begin{pmatrix} s\cdot 2^{-a/2} & 0 \\ 0 & s\cdot 2^{a/2} \end{pmatrix}$ where $\theta \sim \mathcal{U}(-\pi, \pi)$ and $a \sim \mathcal{U}(0,2)$ (note that $\sqrt{\det A} = s$ can still be interpreted as scale).

5 Experiments

Datasets. While we use several datasets in our evaluation (table 1), we mainly focus on HPSequences which builds on the images of HPATCHES [3]. This contains image sequences

[6] We have experimented using region stability as a detection score surrogate, as defined in VLFeat [35], but we did not obtain any consistent results.

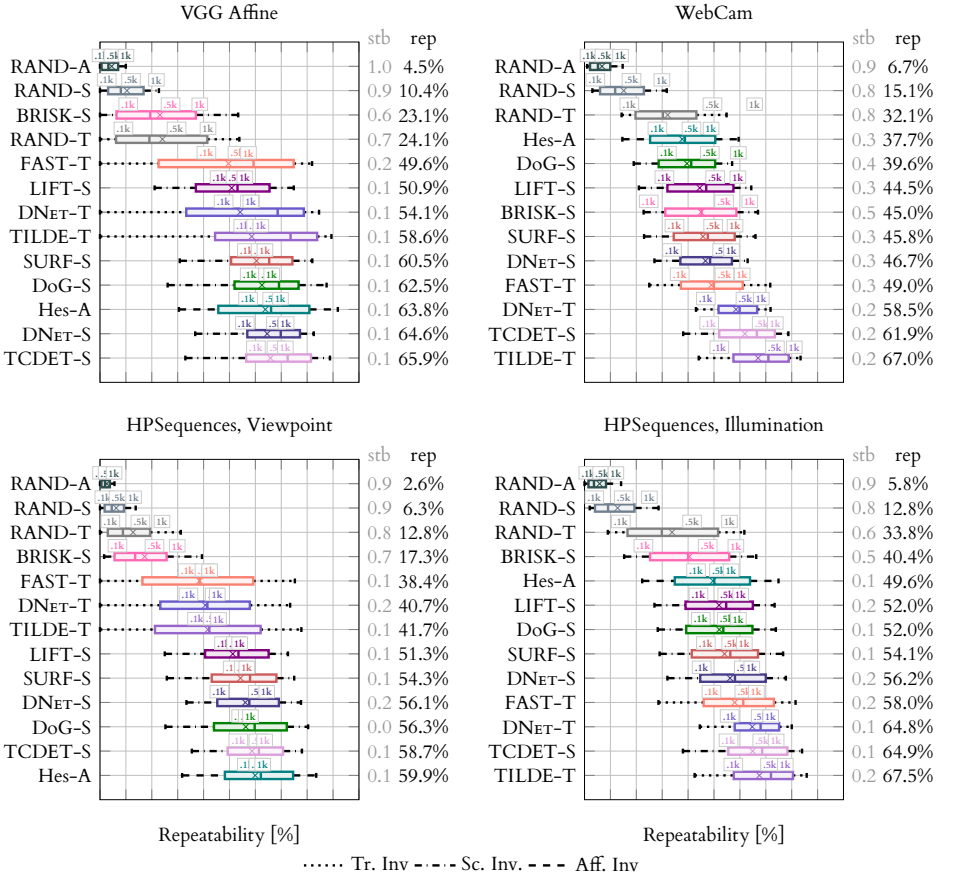

Figure 2: Repeatability of selected detectors on VGG Affine, Webcam, and HPSequences viewpoint/illumination sequences (HP-V vs HP-I). See section 3.3 for the notation.

Table 3: Complete results - average repeatability of the selected detectors on all presented homography datasets.

Det	HP-I [3]			HP-V [3]			HP-I+V [3]			VGG [22]			WEBC [36]			EFOCI [40]			HANN [8]			Avg. *rnk*
	stb	*rep*	*rnk*	*stb*	*rep*	*rnk*	*stb*	*rep*	*rnk*	*stb*	*rep*	*rnk*	*stb*	*rep*	*rnk*	*stb*	*rep*	*rnk*	*stb*	*rep*	*rnk*	
TCDET-S	0.1	**64.91**	2	0.1	**58.71**	2	0.1	**61.76**	1	0.1	**65.85**	1	0.2	**61.92**	2	0.2	**58.25**	1	0.1	51.55	5	2.00
DNET-S	0.2	56.20	5	0.2	56.12	4	0.2	**56.16**	2	0.1	**64.56**	2	0.3	46.70	5	0.2	**53.05**	2	0.1	**54.72**	2	3.14
TILDE-T	0.2	**67.52**	1	0.1	41.67	7	0.1	54.37	4	0.1	58.58	6	0.2	**67.03**	1	0.2	50.53	4	0.1	40.37	8	4.43
Hes-A	0.1	49.60	9	0.1	**59.94**	1	0.1	**54.86**	3	0.1	**63.84**	3	0.3	37.72	10	0.2	47.30	6	0.0	**58.73**	1	4.71
DoG-S	0.1	52.04	7	0.0	**56.29**	3	0.1	54.20	5	0.1	62.53	4	0.4	39.56	9	0.2	**51.12**	3	0.0	**54.71**	3	4.86
SURF-S	0.1	54.05	6	0.1	54.25	5	0.1	54.16	6	0.1	60.45	5	0.3	45.76	6	0.2	49.10	5	0.0	51.76	4	5.29
DNET-T	0.1	**64.79**	3	0.2	40.65	8	0.1	52.51	7	0.1	54.15	7	0.2	**58.47**	3	0.2	46.84	7	0.1	40.82	7	6.00
LIFT-S	0.2	51.96	8	0.1	51.27	6	0.1	51.61	8	0.1	50.85	8	0.3	44.48	8	0.2	45.84	8	0.1	41.20	6	7.43
FAST-T	0.2	57.99	4	0.1	38.43	9	0.1	48.04	9	0.2	49.63	9	0.3	49.01	4	0.2	39.74	9	0.1	39.44	9	7.57
BRISK-S	0.5	40.43	10	0.7	17.25	10	0.5	28.64	10	0.6	23.14	11	0.5	45.03	7	0.6	29.11	10	0.7	12.58	10	9.71
RAND-T	0.6	33.81	11	0.8	12.78	11	0.7	23.11	11	0.7	24.11	10	0.8	32.10	11	0.6	28.09	11	0.8	12.03	11	10.86
RAND-S	0.8	12.78	12	0.9	6.27	12	0.9	9.47	12	0.9	10.41	12	0.8	15.14	12	0.8	17.11	12	0.9	5.61	12	12.00
RAND-A	0.9	5.82	13	0.9	2.59	13	0.9	4.17	13	1.0	4.50	13	0.9	6.74	13	0.9	8.09	13	0.8	2.61	13	13.00

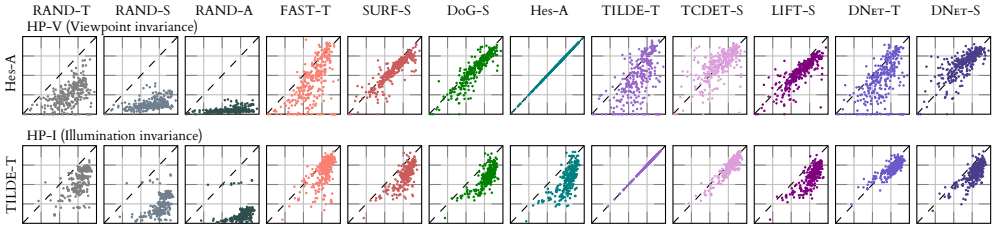

Figure 3: Comparison of repeatability distributions of pair of detectors on different subsets HPSequences (HP-V and HP-I), $n = 1000$. The x-axis is a repeatability of the reference detector (specified by row) and y-axis is repeatability of the selected detector, specified by column. Each point in a plot represents a repeatability of a single image pair.

in a similar format to the original VGG Affine dataset, but with an order of magnitude more sequences, divided in viewpoint (HP-V) and illumination (HP-I) changes.

Aggregated evaluation. We first evaluate the average repeatability of the detectors (fig. 2 and table 3) as defined in eq. (1). For the older **VGG Affine**, translation invariant detectors such as TILDE-T are competitive in median/average with more invariant detectors (-S, -A), but in 10% of the cases fail catastrophically (see the whiskers). The latter problem is solved by scale invariance and the best detectors use Hessian or Laplacian-based scale selection (-S). On the **Webcam** dataset, which contains only illumination changes, RAND-T is surprisingly competitive (mostly due to the fact that scale is always selected consistently), on part with more complex -S and -A detectors. TILDE-S, which is learned on this dataset, is unsurprisingly the winner.

Next, we look at **HPSequences**, starting from the viewpoint sequences (HP-V). Compared to the previous datasets, the RAND-T,S,A baselines perform much worse, confirming that this data is significantly harder. The best detectors are variants of the Hessian one, which is popular in instance retrieval [2, 27], and scale selection (-S) brings in general an advantage; however, the benefits of Baumberg [5] affine adaptation (-A) is small. In general, the top six detectors perform similarly. For the illumination sequences (HP-I, Webcam), since scale does not change, -T detectors are advantaged. The best performance is again achieved with the TILDE-T, which therefore generalises beyond the Webcam dataset.

From the relatively high performance of the RAND-T detector, we can see that it is crucial to compare detectors of similar classes. This also justifies the use of ellipse overlap over the simpler distance of keypoint centres for detector evaluation.

For the stability across detection thresholds (2), we see in table 3 that the majority of the best performing detectors have their stability errors under 10%. However, the stability is much lower for the BRISK and RANDOM detectors, which indicates that the BRISK detection scores are not predictive of the detector performance.

Non-modal performance. A limitation of the analysis above is that it relies on aggregated measures that may hide particular example cases where a given detector has a significant advantage, such as an extreme viewpoint change. To analyse this possibility, in fig. 3 we plot the repeatability of each detector (y-axis) against the one of the best reference detector (Hes-A and TILDE-T) for all images in the viewpoint (HP-V) and illumination (HP-I) sequences. Points above the diagonal mean that the tested detector (column) obtained higher repeatability on a specific image pair than the reference detector (row). Please note that the distribution of the visualised points across y-axis would give us fig. 2. We can se that TILDE-T tends to uniformly dominate other detectors in HP-I, but for HP-V the best overall

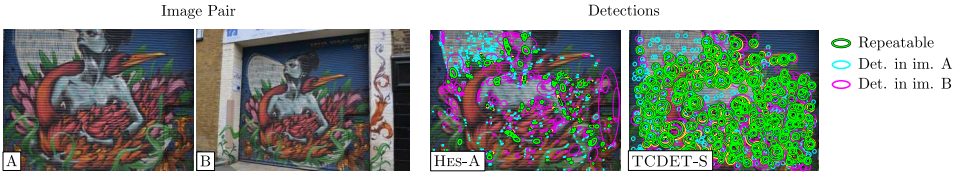

Figure 4: Example of an image pair where a trained detector (rep(TCDET-S) $= 69.8$) achieves better performance compared to a traditional detector (rep(HES-A) $= 28.57$).

detector Hes-A is occasionally outperformed by other detectors such as DNET or TCDET-S, which can therefore be complementary (qualitative example in fig. 4).

Finally, in table 3 we test the consistency of the results across several more datasets (HP-I, -V, -I+V, VGG, Webcam, Edge Foci, Hannover), reporting repeatability, stability and the rank of each detector together with its average rank. Remarkably, detectors learned using the covariance constraint with scale invariance lead the performance across the selected datasets. However traditional detectors generally outperform the trained detectors on tasks where a viewpoint invariance is important. Nonetheless, for learnt detectors this might be mitigated with additional data augmentation or training on datasets with more viewpoint variations. Similarly, the random detectors set a baseline performance for both repeatability and stability across detector's selectivity.

6 Discussion

While learning is poised to change local feature detection, developing a new generation of algorithms almost invariably requires the introduction of improved benchmark datasets. Object detection had PASCAL VOC, deep learning had ImageNet, and handcrafted detectors had VGG Affine. In this paper, we have proposed to improve and extend VGG Affine's protocol to large scale evaluation. While performance of the whole local feature pipeline is important, ability to compare detection performance of different algorithms, without undue influence of the selected description and matching algorithm, is crucial. It not only allows to assess geometric precision of a detector, but in combination with descriptor evaluation it allows to pinpoint the main source of improvement. We are hoping that this detailed analysis will catalyse further progress and advance our understanding of machine learning applied to local feature detection.

Using this benchmark, we have assessed several traditional and deep detectors. We have showed that, while machine learning clearly helps for illumination invariance, for viewpoint invariance traditional methods are still surprisingly competitive. This suggests that there is still significant potential for progress in this area.

Acknowledgements

This research was supported by ERC 677195-IDIU and Programme Grant Seebibyte EP/M013774/1. We would like to thank Dmytro Mishkin for useful feedback on the first versions of this manuscript and Aravindh Mahendran for proof reading. Additionally, we are thankful for the constructive feedback of the BMVC reviewers which helped to improve this work.

References

[1] Henrik Aanæs, Anders Lindbjerg Dahl, and Kim Steenstrup Pedersen. Interesting interest points. *International Journal of Computer Vision*, 97(1):18–35, 2012.

[2] Relja Arandjelović and Andrew Zisserman. Three things everyone should know to improve object retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2911–2918. IEEE, 2012.

[3] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk. HPatch: A benchmark and evaluation of handcrafted and learned local descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[4] Vassileios Balntas, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks. In *Proceedings of the British Machine Vision Conference*, 2016.

[5] A. M. Baumberg. Reliable feature matching across widely separated views. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 774–781, 2000.

[6] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded up robust features. *Proceedings of the European Conference on Computer Vision*, pages 404–417, 2006.

[7] Paul R Beaudet. Rotationally invariant image operators. In *International Joint Conference on Pattern Recognition*, volume 579, page 583, 1978.

[8] Kai Cordes, Bodo Rosenhahn, and Jörn Ostermann. High-resolution feature evaluation benchmark. In *International Conference on Computer Analysis of Images and Patterns*, pages 327–334. Springer, 2013.

[9] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPR Deep Learning for Visual SLAM Workshop*, 2017.

[10] PGT Dias, AA Kassim, and V Srinivasan. A neural network based corner detection method. In *IEEE Int. Conf. on Neural Networks*, 1995.

[11] Philipp Fischer, Alexey Dosovitskiy, and Thomas Brox. Descriptor matching with convolutional neural networks: a comparison to SIFT. *CoRR*, 2014.

[12] Wolfgang Förstner. A feature based correspondence algorithm for image matching. *International Archives of Photogrammetry and Remote Sensing*, 26(3):150–166, 1986.

[13] C. Harris and M. Stephens. A combined corner and edge detector. In *Proc. of The Fourth Alvey Vision Conference*, pages 147–151, 1988.

[14] S. Holzer, J. Shotton, and P. Kohli. Learning to efficiently detect repeatable interest points in depth data. In *Proceedings of the European Conference on Computer Vision*, 2012.

[15] K. Lenc and A. Vedaldi. Learning covariant feature detectors. In *ECCV Workshop on Geometry Meets Deep Learning*, 2016.

[16] Stefan Leutenegger, Margarita Chli, and Roland Y. Siegwart. BRISK: Binary robust invariant scalable keypoints. In *Proceedings of the International Conference on Computer Vision*, pages 2548–2555. IEEE Computer Society, 2011.

[17] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2(60):91–110, 2004.

[18] J. Matas, S. Obdrzálek, and O. Chum. Local affine frames for wide-baseline stereo. In *Proceedings of the International Conference on Pattern Recognition*, 2002.

[19] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *Proceedings of the International Conference on Computer Vision*, 2001.

[20] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *Proceedings of the European Conference on Computer Vision*, pages 128–142. Springer-Verlag, 2002.

[21] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10): 1615–1630, 2005.

[22] Krystian Mikolajczyk, Tinne Tuytelaars, Cordelia Schmid, Andrew Zisserman, Jiri Matas, Frederik Schaffalitzky, Timor Kadir, and Luc Van Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1-2):43–72, 2005.

[23] Dmytro Mishkin, Jiri Matas, Michal Perdoch, and Karel Lenc. Wxbs: Wide baseline stereo generalizations. In *Proceedings of the British Machine Vision Conference*, pages 12.1–12.12. BMVA Press, 2015.

[24] Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Learning discriminative affine regions via discriminability. *CoRR*, abs/1711.06704, 2017.

[25] G. Olague and L. Trujillo. Evolutionary-computer-assisted design of image operators that detect interest points using genetic programming. *Image and Vision Computing*, 2011.

[26] Yuki Ono, Eduard Trulls, Pascal Fua, and Kwang Moo Yi. LF-Net: learning local features from images. *CoRR*, abs/1805.09662, 2018.

[27] Michal Perd'och, Ondrej Chum, and Jiri Matas. Efficient representation of local geometry for large scale object retrieval. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 9–16. IEEE, 2009.

[28] E. Rosten, R. Porter, and T. Drummond. Faster and better: a machine learning approach to corner detection. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010.

[29] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *Proceedings of the European Conference on Computer Vision*, 2006.

[30] Nikolay Savinov, Akihito Seki, Lubor Ladicky, Torsten Sattler, and Marc Pollefeys. Quad-networks: unsupervised learning to rank for interest point detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[31] Johannes Lutz Schönberger, Hans Hardmeier, Torsten Sattler, and Marc Pollefeys. Comparative evaluation of hand-crafted and learned local features. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[32] J. Sochman and J. Matas. Learning fast emulators of binary decision processes. *International Journal of Computer Vision*, 2009.

[33] L. Trujillo and G. Olague. Synthesis of interest point detectors through genetic programming. In *Proc. of GECCO*, 2006.

[34] Tinne Tuytelaars and Luc Van Gool. Matching widely separated views based on affine invariant regions. *International Journal of Computer Vision*, 59(1):61–85, 2004.

[35] A. Vedaldi and B. Fulkerson. VLFeat – An open and portable library of computer vision algorithms. In *Proc. ACM Int. Conf. on Multimedia*, 2010.

[36] Yannick Verdie, Kwang Moo Yi, Pascal Fua, and Vincent Lepetit. TILDE: A Temporally Invariant Learned DEtector. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[37] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. LIFT: Learned Invariant Feature Transform. In *Proceedings of the European Conference on Computer Vision*, 2016.

[38] Kwang Moo Yi, Yannick Verdie, Pascal Fua, and Vincent Lepetit. Learning to Assign Orientations to Feature Points. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[39] Xu Zhang, Felix X Yu, Svebor Karaman, and Shih-Fu Chang. Learning discriminative and transformation covariant local feature detectors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6818–6826, 2017.

[40] C Lawrence Zitnick and Krishnan Ramnath. Edge foci interest points. In *Proceedings of the International Conference on Computer Vision*, pages 359–366. IEEE, 2011.

[41] Larry Zitnick and Piotr Dollar. Edge boxes: Locating object proposals from edges. In *Proceedings of the European Conference on Computer Vision*, 2014.

[42] M. Zuliani, C. Kenney, and B. S. Manjunath. A mathematical comparison of point detectors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005.