MORB: A MULTI-SCALE BINARY DESCRIPTOR

Alessio Xompero[1] *Oswald Lanz*[2] *Andrea Cavallaro*[1]

[1]Centre for Intelligent Sensing, Queen Mary University of London, UK
[2]Fondazione Bruno Kessler, Trento, Italy

ABSTRACT

Local image features play an important role in matching images under different geometric and photometric transformations. However, as the scale difference across views increases, the matching performance may considerably decrease. To address this problem we propose MORB, a multi-scale binary descriptor that is based on ORB and that improves the accuracy of feature matching under scale changes. MORB describes an image patch at different scales using an oriented sampling pattern of intensity comparisons in a predefined set of pixel pairs. We also propose a matching strategy that estimates the cross-scale match between MORB descriptors across views. Experiments show that MORB outperforms state-of-the-art binary descriptors under several transformations.

Index Terms— Local features, Feature matching, Binary descriptor, Multi-scale, MORB

1. INTRODUCTION

Descriptive local image features are fundamental for a number of applications, including Structure from Motion [1], Visual SLAM [2], Image Matching and Object Retrieval [3]. A local feature describes the neighbourhood of a keypoint that was localised by a detector, such as Harris' [4] or Scale Invariant Feature Transform (SIFT) [3].

Multi-scale detectors aim to achieve scale invariance across views [5]. Detectors can localise keypoints for each scale independently and then select those with maximum response across scales [6] or directly in the scale-space domain [3]. While local features are usually described at the scale determined by the detector [3][7][8], descriptors can also be extracted at different scales and reduced by approximation [9], pooling [10] or masking [11].

Local features can be hand-crafted or learnt. Examples of *hand-crafted features* include SIFT [3], Speeded Up Robust Features (SURF) [12] and Binary Robust Invariant Scale Key-point (BRISK) [8]. Hand-crafted features for real-time applications include binary descriptors generated by tests that compare intensity values of pixel pairs [13][8][14] or small window triplets [15] within the neighbourhood (or patch) of a keypoint. Keypoints can be detected with, for example, Features from Accelerated Segment Test (FAST) [16] or Adaptive and Generic Accelerated Segment Test (AGAST) [17]. The Binary Robust Invariant Elementary Features (BRIEF) descriptor randomly samples the tests from a Gaussian distribution [13]. The BRISK descriptor [8] uses a deterministic sampling pattern whose points lie on appropriately scaled concentric circles. The Fast REtinA Keypoint (FREAK) descriptor [14] uses a circular pattern with higher density near the centre of the keypoint.

Examples of *learnt features* include DeepDesc [18], DeepCompare [19], TFeat [20] DeepBit [21] and Learned Invariant Feature Transform (LIFT) [22]. These methods exploit Convolutional Neural Networks and model objective functions to discriminate correct and incorrect matches learnt during training with ground-truth data. DeepBit learns instead a binary descriptor in an unsupervised manner. Learnt descriptors have however not yet outperformed hand-crafted features [23]. A somehow hybrid approach is that of Oriented FAST and Rotated BRIEF (ORB) [7], a binary descriptor built on the FAST detector [16], the BRIEF descriptor [13], and a learnt sampling pattern of pixel pairs. However, the matching performance of ORB decreases considerably when the scale difference across views increases.

To address this problem, we propose MORB, a multi-scale binary descriptor that can cope with large scale-variations between views. The proposed descriptor concatenates binary ORB descriptors extracted at multiple scales. These descriptors are appropriately rotated to adapt to the varying content within a patch at different scales (see Fig. 1). To identify the best match and the scale difference among images, we then compute the cross-scale distance between MORB descriptors of each view. Correct matches can be identified at descriptor scales that differ from the scale of the keypoint.

This paper is organised as follows. Section 2 introduces the multi-scale binary descriptor. Section 3 describes the associated cross-scale matching strategy. Section 4 discusses the experimental results. Finally, in Section 5 we draw the conclusion.

2. MULTI-SCALE ORB

Let $\mathcal{I} = \{I_s\}_{s=1}^{S}$ be a Gaussian pyramid of image I, where each layer I_s is recursively down-sampled by a factor λ, up to scale S. We apply in each I_s independently the FAST detector [16] and retain only the F features across scales with the highest Harris[1] response [4] in an adaptive way:

$$F_s = \begin{cases} \frac{1-\lambda}{1-\lambda^S}, & \text{if } s = 1, \\ \lambda F_{s-1}, & \text{if } 1 < s < S, \\ \max\left(F - F_{s-1}, 0\right) & \text{if } s = S, \end{cases} \tag{1}$$

where F_s is the number of features for each scale s.

After smoothing each layer I_s with a 2D Gaussian filter with size $W = 7$ and standard deviation $\sigma = 2$, we extract the descriptor $\mathbf{d}_{\mathbf{p},s}^{m}$ using the rotated ORB sampling pattern on a $G \times G$ patch $\mathbf{p}$ centred at each feature location:

$$\mathbf{d}_{\mathbf{p},s}^{m} = [\tau_{\mathbf{p},s}(\mathbf{u}_1, \mathbf{v}_1), ..., \tau_{\mathbf{p},s}(\mathbf{u}_q, \mathbf{v}_q), ...\tau_{\mathbf{p},s}(\mathbf{u}_{256}, \mathbf{v}_{256})], \tag{2}$$

where $\mathbf{u}_q$ and $\mathbf{v}_q$ are the positions of each pixel pair in the sampling pattern, and $m = 1, \dots, F$ is the index of the m-th feature.

[1] The Harris score is preferable to the FAST score as cornerness measure [7].

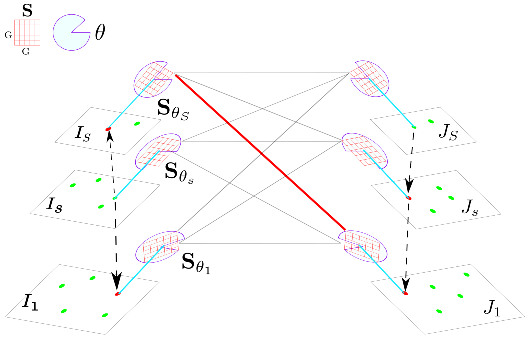


Fig. 1. The MORB multi-scale descriptor and its cross-scale matching. Once a keypoint is detected at a scale s (green dot), MORB samples its location for each layer of an image pyramid (red dots) and determines the patch orientation. A rotated descriptor based on a sampling pattern for binary derivatives is extracted for each scale and then contributes to the MORB descriptor. The matching across scales between MORBs from different view points determines the scale difference.

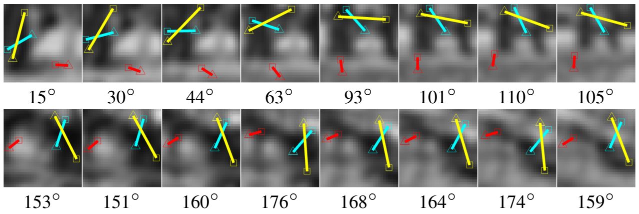


Fig. 2. Sample patch orientation changes along the scales (from left to right) and across views (top row: view 1; bottom row: view 2) for the proposed MORB descriptor. For each patch we show its orientation in degrees and 3 sample rods (red, cyan, yellow) from the ORB sampling pattern.

The function $\tau_{\mathbf{p},s}(\cdot,\cdot)$ is a binary test on the intensity values of pixel pairs at scale s:

$$\tau_{\mathbf{p},s}(\mathbf{u}_q,\mathbf{v}_q) = \begin{cases} 1, & \text{if} \quad I_{\mathbf{p},s}(\mathbf{u}_q) < I_{\mathbf{p},s}(\mathbf{v}_q), \\ 0, & \text{otherwise}, \end{cases} \tag{3}$$

where $I_{\mathbf{p},s}(\mathbf{u}_q)$ and $I_{\mathbf{p},s}(\mathbf{v}_q)$ are the intensity values in patch $\mathbf{p}$ at pixel position $\mathbf{u}_q$ and $\mathbf{v}_q$, respectively, and $q = 1, ..., 256$ corresponds to positions defined by the ORB sampling pattern $\mathbf{S}$. The pattern $\mathbf{S}$ consists of learnt pixel pairs with high variance and low correlation in their binary derivative [7].

As scale variation is already contained in the Gaussian pyramid, we keep the patch size fixed across scales. This changes the portion of the scene captured by the patch at different scales. We also recompute the orientation angle θ_s for each scale s. The angle θ_s is calculated with respect to the centre of mass of the patch defined by the intensity centroid [24]. Each $\mathbf{d}^m_{\mathbf{p},s}$ is then extracted using the rotated pattern $\tilde{\mathbf{S}}_s$, after the rotation $\mathbf{R}_{\theta_s} \in SO(2)$ is applied to $\mathbf{S}$: $\tilde{\mathbf{S}}_s = \mathbf{R}_{\theta_s}\mathbf{S}$. Fig. 2 is an example of rotated pattern at different scales.

The MORB descriptor $\mathbf{d}^m_{\mathbf{p}}$ of patch $\mathbf{p}$ concatenates patch descriptors extracted at all layers of the image pyramid:

$$\mathbf{d}^m_{\mathbf{p}} = \left[\mathbf{d}^m_{\mathbf{p},1}, \ldots, \mathbf{d}^m_{\mathbf{p},S}\right] \tag{4}$$

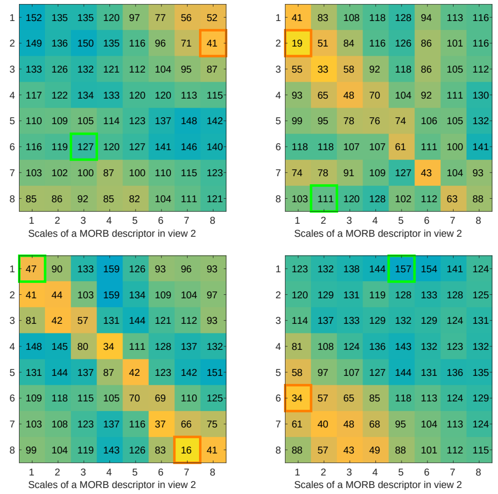


Fig. 3. Hamming distance matrices for a sample of four pairs of MORB descriptors. Green boxes denote the scales of keypoint detections. Orange boxes denote the scales of the minimum Hamming distance (of correct multi-view matches). Note the difference between the scales of keypoint detection and descriptor matching.

and can support feature matching across views with significant scale change.

However, to extract the multi-scale descriptor for each keypoint, MORB scales its image coordinates for each layer s and approximates them by rounding. This can result in keypoints whose distance to the image border at the coarsest scale of the Gaussian pyramid after scaling is lower than half of the patch size G and thus inhibits the extraction of the multi-scale descriptor. We therefore discard these keypoints that are so close to the borders. We also remove duplicates by discarding one keypoint from every pair of keypoints that are at most 2 pixels from each other when up-sampled to the original image scale.

3. CROSS-SCALE MATCHING

Let $\mathbf{d}^m_{\mathbf{p}}$ and $\mathbf{d}^n_{\mathbf{p}}$ be multi-scale descriptors of a keypoint m in a view and a keypoint n in another view, respectively.

Matching strategies such as nearest neighbour [25] or bag-of-words [26] are not directly applicable to the MORB descriptor, as the scales where two local features can be matched are unknown. We therefore aim to identify the minimum cross-scale distance for each feature pair.

We first compute an all-to-all single descriptor distance across scales between each $\mathbf{d}^m_{\mathbf{p},s}$ and $\mathbf{d}^n_{\mathbf{p},l}$, and then we take the minimum as the cross-scale distance between the keypoints:

$$h_{m,n} = \min_{s,l \in \mathcal{S}} \mathbf{d}^m_{\mathbf{p},s} \oplus \mathbf{d}^n_{\mathbf{p},l}, \tag{5}$$

where $\mathcal{S} = \{1, \ldots, S\}$, $\oplus$ is the XOR operator and $\mathbf{d}^m_{\mathbf{p},s} \oplus \mathbf{d}^n_{\mathbf{p},l}$ is the Hamming distance between two ORB descriptors. The scales where the minimum match is found, s^* and l^*, determine the scale offset between the two keypoints $(s^* - l^*)$.

Fig. 3 shows examples of four cross-scale Hamming distance matrices between matched MORB descriptors. Fig. 4 shows an example of cross-scale matching, where the match occurs at scales that are different from the detection scales.

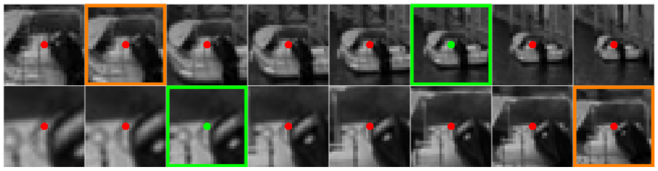


Fig. 4. Sample corresponding patches at multiple scales across views with considerable scale variation (top row: view 1; bottom row: view 2). Note the difference between the scales of the keypoint detections (green squares) and of the MORB matching (orange squares). This case is related to the top-left matrix in Fig. 3.

The set of putative matches $\mathcal{V}$ is estimated via nearest neighbour [25] and with a threshold τ on the descriptor distance to separate true and false positive putative matches. While the distribution of false positives can lie on high descriptor distances, the distribution of correct matches covers the low ones [13]. We obtain a set of matches between two views as

$$\mathcal{N} = \left\{ (m^*, n) \mid m^* = \operatorname*{arg\,min}_{m\in\mathcal{F}} h_{m,n}, n \in \mathcal{F}, h_{m,n} \le \tau \right\}, \quad (6)$$

where $\mathcal{F} = \{1, \ldots, F\}$. Similarly, we obtain the set of reverse matches as

$$\mathcal{M} = \left\{ (m, n^*) \mid n^* = \operatorname*{arg\,min}_{n\in\mathcal{F}} h_{m,n}, m \in \mathcal{F}, h_{m,n} \le \tau \right\}. \quad (7)$$

The set of valid matches is then $\mathcal{V} = \mathcal{N} \cap \mathcal{M}$. In the next section we analyse the impact of the threshold on the effectiveness of our approach.

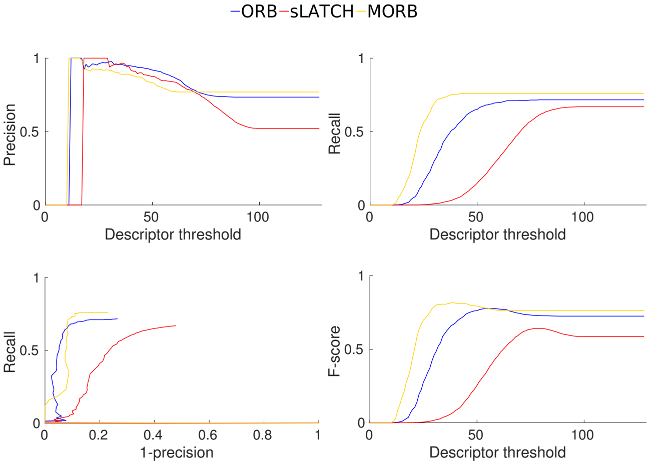
4. EXPERIMENTS

We compare MORB with ORB [7] (OpenCV 3.3 implementation) and with Learned Arrangements of Three patCH codes (LATCH) [15] using keypoints detected with MORB. In this case, we refer to ORB and LATCH as cORB and oLATCH, respectively. As LATCH was paired with SIFT in [15], we also report the results of LATCH applied on keypoints detected with SIFT (sLATCH). Furthermore, we report results of ORB with its own detections and we test an all-to-all matching of independent ORB descriptors extracted for all scales (ORB-ALL).

In the detection phase, MORB uses the same approach as ORB and thus we consider the same settings: the FAST threshold is 20, the patch size is $G = 31$, the number of scales is $S = 8$, and the scale factor is $\frac{1}{\lambda} = 1.2$. Even if the default target number of features F for each image is 500 in ORB, considering recent evaluations we set $F = 1000$ [22]. (We also analysed the performance of MORB and ORB by varying F from 500 to 1500 with step 250, but we did not observe any significant performance changes.)

We use as dataset the Oxford Affine Covariance Regions Dataset (Oxford ACRD) [25] that consists of eight sets of six images under five different conditions: in-plane rotation changes and scale changes (*bark* and *boat*), viewpoint changes (*graf* and *wall*), image blur (*bikes* and *trees*), illumination changes (*leuven*), and JPEG compression (*ubc*). Moreover, we consider the *venice* set from [27] to evaluate performance under scale variations only.

As we propose a scale-aware nearest neighbour matching strategy for MORB, we evaluate ORB and LATCH with the nearest neighbour approach as *similarity matching* [25]. We define a correspondence (as well as a correct match) as the pair of keypoints with the lowest distance below 2.5 pixels after homography transformation (homographies are provided as ground-truth along with the dataset), with all keypoints scaled up to the original scale, as suggested in [27]. To analyse the impact of the descriptor threshold, we vary τ from 0 to 128, (*i.e.* half of the size of the descriptor) and we then compute the number of matches $\mathcal{V}$ and the corresponding number of correct matches to generate precision and recall curves. Moreover, the area under the curve can be used to compare methods [10] [15] [28].



	ORB	sLATCH	MORB
Precision	**.76**	.63	.75
Recall	.53	.35	**.63**
Recall vs 1-precision	.15	**.19**	.13
F-score	.58	.36	**.66**

Fig. 5. Precision, recall, recall vs 1-precision, and F-score curves for the image pair *boat* 1 – 2. It can be noted in the table that using area under the recall vs 1-precision curve can lead to inconsistent ranking. Area under the F-score curve better preserves precision and recall behaviour as it is computed from their harmonic mean.

Precision and recall can be analysed together through recall vs 1-precision curves [25] or the F-score$= 2\frac{\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}$. Here, we propose to evaluate the methods with the area under the F-score curve as we observed that computing the area under the recall vs. 1-precision curves with the nearest neighbour matching strategy can lead to a method ranking that is inconsistent with the ranking obtained with more detailed area under the precision or recall curves (see Fig. 5). In the recall vs 1-precision curve, a good method should not significantly decrease in precision and should keep a high recall, or keep a high recall even if the precision tends to zero. However, good methods in precision and recall may cover a smaller area than methods decreasing in precision and having a lower recall, thus resulting in lower performance. On the other hand, the F-score can preserve the performance of precision and recall for evaluating the methods. We therefore refer to the area under the F-score curve as Nearest Neighbour Average F-score (NN–AF). As the NN–AF is insufficient to compare methods, we also compute the matching score (MS), *i.e.* the number of correct matches over the minimum number of features in common after homography transformation, with $\tau = 128$.

Table 1 shows the NN–AF and MS results for each image pair in *venice* and in each set of the Oxford ACRD dataset. MORB

Table 1. Nearest Neighbour Average F-score (NN–AF) and Matching Score (MS) for each image pair for each set of images.

		NN–AF						MS					
		ORB	sLATCH	cORB	oLATCH	ORB-ALL	MORB	ORB	sLATCH	cORB	oLATCH	ORB-ALL	MORB
venice	1 – 2	.70	.42	.57	.35	.62	**.73**	**.57**	.45	.46	.41	.43	.51
	1 – 3	.61	.04	.45	.05	.49	**.69**	**.45**	.05	.33	.06	.30	.44
	1 – 4	.41	.00	.31	.00	.28	**.56**	.20	.00	.16	.00	.12	**.25**
	1 – 5	.35	.00	.25	.00	.17	**.38**	.12	.00	.09	.00	.05	**.14**
	1 – 6	.12	.00	.14	.00	.05	**.14**	.03	.00	.04	.00	.01	**.05**
	1 – 7	.01	.00	.02	.00	.02	**.09**	.00	.00	.01	.00	.00	**.03**
bark	1 – 2	.28	.14	.22	.12	.25	**.33**	.11	**.12**	.09	.08	.09	**.12**
	1 – 3	.09	.00	.09	.00	.08	**.15**	.03	.00	.02	.00	.02	**.04**
	1 – 4	.18	.00	.13	.00	.16	**.37**	.04	.00	.03	.00	.04	**.10**
	1 – 5	.13	.00	.11	.00	.12	**.37**	.03	.00	.03	.00	.03	**.10**
	1 – 6	.04	.00	.04	.00	.02	**.09**	.01	.00	.01	.00	.00	**.02**
boat	1 – 2	.58	.36	.50	.40	.50	**.66**	.46	.31	.41	.41	.36	**.48**
	1 – 3	.53	.06	.46	.11	.44	**.66**	.39	.06	.34	.14	.28	**.43**
	1 – 4	.36	.00	.32	.00	.28	**.51**	.22	.00	.20	.00	.15	**.30**
	1 – 5	.27	.00	.28	.00	.19	**.46**	.15	.00	.15	.00	.09	**.24**
	1 – 6	.08	.00	.09	.00	.05	**.16**	.04	.00	.04	.00	.02	**.07**
graffiti	1 – 2	.55	.43	.49	.33	.48	**.64**	**.46**	.39	.39	.33	.34	.45
	1 – 3	.27	.09	.21	.12	.21	**.33**	.20	.09	.16	.12	.14	**.23**
	1 – 4	.11	.02	.10	.03	.08	**.12**	**.08**	.02	.07	.04	.05	**.08**
	1 – 5	**.02**	.00	**.02**	.00	.01	.01	**.02**	.00	**.02**	.00	.01	.01
	1 – 6	.00	.00	**.01**	**.01**	.00	.00	.00	.00	**.01**	**.01**	.00	.00
wall	1 – 2	.48	.50	.44	.34	.46	**.65**	.36	**.50**	.33	.31	.32	.45
	1 – 3	.44	.35	.38	.26	.39	**.61**	.34	.34	.30	.26	.29	**.44**
	1 – 4	.23	.16	.22	.13	.21	**.36**	.14	.14	.14	.11	.12	**.21**
	1 – 5	.08	.04	.08	.04	.07	**.13**	.04	.04	.04	.03	.03	**.07**
	1 – 6	**.01**	.00	.00	**.01**	**.01**	**.01**	.00	.00	.00	.00	.00	**.01**
bikes	1 – 2	.71	.66	.62	.62	.66	**.76**	**.61**	.50	.51	.53	.49	.55
	1 – 3	.65	.63	.56	.58	.61	**.73**	**.54**	.52	.45	.47	.44	.50
	1 – 4	.53	.55	.44	.45	.55	**.67**	.38	**.46**	.31	.35	.35	.41
	1 – 5	.43	.51	.32	.34	.46	**.57**	.28	**.44**	.22	.26	.28	.34
	1 – 6	.35	.41	.25	.26	.37	**.48**	.20	**.36**	.15	.18	.20	.25
trees	1 – 2	.49	.30	.39	.36	.47	**.59**	.32	.21	.25	.26	.28	**.34**
	1 – 3	.41	.24	.33	.30	.40	**.55**	.22	.18	.18	.18	.21	**.27**
	1 – 4	.27	.15	.21	.23	.25	**.35**	.12	.10	.10	.13	.11	**.15**
	1 – 5	.21	.12	.16	.16	.23	**.30**	.09	.09	.07	.09	.09	**.11**
	1 – 6	.13	.07	.11	.13	.15	**.22**	.04	.06	.04	.06	.05	**.08**
leuven	1 – 2	.67	**.77**	.61	.59	.57	.70	.48	**.59**	.43	.42	.37	.44
	1 – 3	.60	**.73**	.54	.54	.52	.63	.38	**.55**	.36	.36	.31	.37
	1 – 4	.55	**.68**	.47	.51	.48	.62	.33	**.52**	.29	.31	.26	.34
	1 – 5	.51	**.64**	.41	.47	.45	.57	.28	**.48**	.24	.26	.23	.30
	1 – 6	.46	**.58**	.41	.44	.42	.53	.26	.43	.24	.26	.22	**.28**
ubc	1 – 2	**.93**	.76	.90	.88	.91	.89	**.90**	.57	.86	.86	.84	.77
	1 – 3	**.90**	.64	.86	.83	.87	.89	**.84**	.48	.79	.79	.77	.74
	1 – 4	.84	.50	.77	.74	.82	**.87**	**.78**	.35	.71	.72	.71	.71
	1 – 5	.70	.35	.63	.58	.69	**.79**	**.63**	.20	.57	.58	.58	**.63**
	1 – 6	.57	.26	.50	.45	.56	**.66**	**.51**	.17	.45	.46	.44	.48
ACRD avg.		.39	.29	.34	.28	.36	**.47**	.28	.23	.25	.23	.24	**.30**
Total avg.		.39	.27	.34	.26	.35	**.47**	.28	.21	.24	.21	.23	**.29**

outperforms the other descriptors in the three sets with either only scale variations (*venice*) or in-plane rotations and scale variations (*bark* and *boat*) as well as in other sets under other geometric and photometric transformations, except for illumination changes (*leuven*). In these last cases sLATCH is the best performing method. As oLATCH performs similarly to cORB in *leuven*, the good performance of sLATCH is possibly due to the keypoint detected with SIFT. Nevertheless, LATCH is sensitive to scale changes both with the ORB and the SIFT detector. Most of the NN–AF performance are supported by a similar or higher MS, showing the capability of MORB to find more correct matches than the other descriptors. We can also observe that cORB performs worse than ORB. This performance can be caused by the discarded keypoints that could be relevant for the matching. We proved the effectiveness of our cross-scale matching over ORB-ALL showing that the independence assumption of single descriptors across scales for each feature decreases the matching performance.

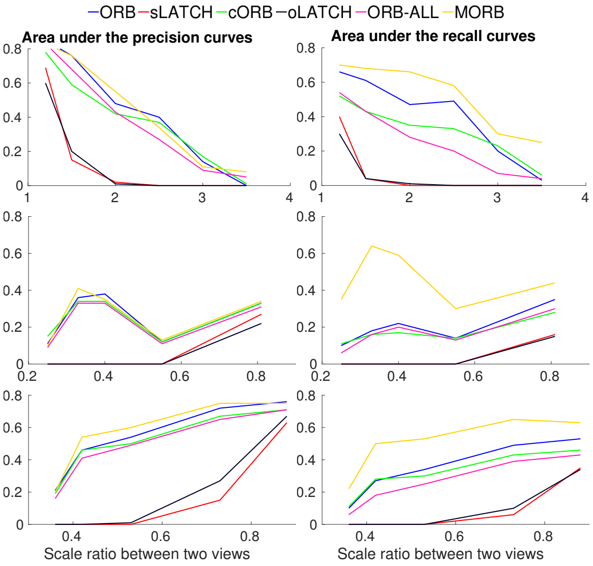


Fig. 6. Area under the precision curves (left) and area under the recall curves (right) when increasing the scale ratio between image pairs in *venice* (top), *bark* (middle) and *boat* (bottom). While *venice* shows an increasing zoom in, *bark* and *boat* shows an increasing zoom out of the target images with respect to the reference image.

Fig. 6 shows the area under the precision curves and the area under the recall curves in relation to the scale ratio between the image pairs in *venice*, *boat* and *bark*. While all ORB variants and MORB have similar precision performance, MORB outperforms in recall, thus estimating more correct matches than the other descriptors. As mentioned earlier, sLATCH and oLATCH perform poorly except when the scale change is small (scale ratio close to 1) where their performance is closer to that of ORB.

5. CONCLUSIONS

We proposed MORB, a binary descriptor that uses multiple scales of a Gaussian pyramid to increase matching accuracy under scale changes. We also proposed a scale-aware nearest neighbour matching strategy that estimates the minimum cross-scale distance between two MORB descriptors and, as by-product, can infer the scale ratio between pairs of local features. The matched scales tend to differ from the scales where the keypoints were localised: this leads to an increase in the number of correct matches and to a better performance than ORB, which considers the scale of the detection only.

While the proposed method is based on ORB [7], the overall pipeline is modular and can be generalized to other (binary) keypoint descriptors.

6. REFERENCES

[1] J. L. Schönberger and J. Frahm, "Structure-from-motion revisited," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 27–30 June 2016.

[2] R. Mur-Artal, J.M.M. Montiel, and J.D. Tardos, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.

[3] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[4] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. of the fourth Alvey Vision Conference*, Manchester, UK, 31 Aug./2 Sept. 1988.

[5] T. Tuytelaars, K. Mikolajczyk, et al., "Local invariant feature detectors: a survey," *Foundations and Trends® in Computer Graphics and Vision*, vol. 3, no. 3, pp. 177–280, Jan. 2008.

[6] K. Mikolajczyk and C. Schmid, "Indexing based on scale invariant interest points," in *Proc. of the IEEE International Conference on Computer Vision*, Vancouver, Canada, 7–14 July 2001.

[7] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: an efficient alternative to SIFT or SURF," in *Proc. of the IEEE International Conference on Computer Vision*, Barcelona, Spain, 6–13 Nov. 2011.

[8] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary Robust Invariant Scalable Keypoints," in *Proc. of the IEEE International Conference on Computer Vision*, Barcelona, Spain, 6–13 Nov. 2011.

[9] T. Hassner, V. Mayzels, and L. Zelnik-Manor, "On SIFTs and their Scales," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Providence, RI, USA, 16–21 June 2012.

[10] J. Dong and S. Soatto, "Domain-size pooling in local descriptors: DSP-SIFT," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Boston, MA, USA, 7–12 June 2015.

[11] V. Balntas, L. Tang, and K. Mikolajczyk, "Binary Online Learned Descriptors," *IEEE Tran. on Pattern Analysis and Machine Intelligence*, vol. PP, no. 99, Mar. 2017.

[12] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded Up Robust Features," in *Proc. of the European Conference on Computer Vision*, Graz, Austria, 7–13 May 2006.

[13] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary Robust Independent Elementary Features," in *Proc. of the European Conference on Computer Vision*, Heraklion, Crete, Greece, 5–11 Sept. 2010.

[14] A. Alahi, R. Ortiz, and P. Vandergheynst, "FREAK: Fast Retina Keypoint," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Providence, RI, USA, 16–21 June 2012.

[15] G. Levi and T. Hassner, "LATCH: Learned Arrangements of Three Patch Codes," in *Proc. of the IEEE Winter Conference on Applications of Computer Vision*, Lake Placid, NY, USA, 7–10 Mar. 2016.

[16] E. Rosten and T. Drummond, "Machine Learning for High-Speed Corner Detection," in *Proc. of the European Conference on Computer Vision*, Graz, Austria, 7–13 May 2006.

[17] E. Mair, G. D. Hager, D. Burschka, M. Suppa, and G. Hirzinger, "Adaptive and generic corner detection based on the accelerated segment test," in *Proc. of the European Conference on Computer Vision*, Heraklion, Crete, Greece, 5–11 Sept. 2010.

[18] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer, "Discriminative learning of deep convolutional feature point descriptors," in *Proc. of the IEEE International Conference on Computer Vision*, Santiago, Chile, 7–13 Dec. 2015.

[19] S. Zagoruyko and N. Komodakis, "Learning to compare image patches via convolutional neural networks," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Boston, MA, USA, 7–12 June 2015.

[20] V. Balntas, E. Riba, D. Ponsa, and K. Mikolajczyk, "Learning local feature descriptors with triplets and shallow convolutional neural networks.," in *Proc. of The British Machine Vision Conference*, York, UK, 19–22 Sept. 2016.

[21] K. Lin, J. Lu, C. Chen, and J. Zhou, "Learning compact binary descriptors with unsupervised deep neural networks," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 27–30 June 2016.

[22] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua, "LIFT: Learned Invariant Feature Transform," in *Proc. of the European Conference on Computer Vision*, Amsterdam, The Netherlands, 8–16 Oct. 2016.

[23] J. L. Schönberger, H. Hardmeier, T. Sattler, and M. Pollefeys, "Comparative evaluation of hand-crafted and learned local features," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, 21–26 July 2017.

[24] P. L. Rosin, "Measuring corner properties," *Computer Vision and Image Understanding*, vol. 73, no. 2, pp. 291–307, Feb. 1999.

[25] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Tran. on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, Oct. 2005.

[26] D. Gálvez-López and J. D. Tardos, "Bags of Binary Words for Fast Place Recognition in Image Sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, 2012.

[27] J. Heinly, E. Dunn, and J. Frahm, "Comparative evaluation of binary features," in *Proc. of the European Conference on Computer Vision*, Firenze, Italy, 7–13 Oct. 2012.

[28] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk, "HPatches: A benchmark and evaluation of handcrafted and learned local descriptors," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, 21–26 July 2017.