Self-Supervised Learning of Image Scale and Orientation

Jongmin Lee
ljm1121@postech.ac.kr
Yoonwoo Jeong
jeongyw12382@postech.ac.kr
Minsu Cho
mscho@postech.ac.kr

Computer Vision Lab.
POSTECH
Pohang, Republic of Korea

Abstract

We study the problem of learning to assign a characteristic pose, i.e., scale and orientation, for an image region of interest. Despite its apparent simplicity, the problem is non-trivial; it is hard to obtain a large-scale set of image regions with explicit pose annotations that a model directly learns from. To tackle the issue, we propose a self-supervised learning framework with a histogram alignment technique. It generates pairs of image patches by random rescaling/rotating and then train an estimator to predict their scale/orientation values so that their relative difference is consistent with the rescaling/rotating used. The estimator learns to predict a non-parametric histogram distribution of scale/orientation without any supervision. Experiments show that it significantly outperforms previous methods in scale/orientation estimation and also improves image matching and 6 DoF camera pose estimation by incorporating our patch poses into a matching process.

1 Introduction

Local feature representation lies at the heart of computer vision, and extensive research has been conducted on detecting and/or describing local features [3, 9, 22, 38]. With the remarkable advance of convolutional neural networks (CNNs) [15, 43, 46], the dense feature map output of convolutional layers has largely replaced the classic hand-crafted feature representation in a wide range of tasks [2, 8, 9, 18, 34, 35]. However, since the convolutional feature map is equivariant only to translation but not to the other common pose variations, *e.g.*, scaling and rotating, assigning a characteristic pose of an image or region of interest is required to extract an accurate descriptor for many vision problems such as visual correspondence, registration, retrieval, localization, and 3D reconstruction [1, 21, 22, 30, 33, 39, 40, 41]; *e.g.*, the characteristic scale and/or orientation can be used to extract pose-normalized features from images with different viewpoints or object poses.

Despite its apparent simplicity, the problem of learning to assign a characteristic pose, *i.e.*, scale and orientation, for an image region is non-trivial; it is hard to obtain a large-scale set of image regions with explicit pose annotations that a model directly learns from. To tackle the issue, recent methods [29, 31, 42, 49, 50] use an implicit learning approach

with a surrogate objective where they treat scale and/or orientation as a latent variable; they indirectly train a pose regressor by maximizing the similarity between image regions that are aligned using the estimated pose values.

In this paper, we propose a self-supervised explicit learning framework via a histogram alignment technique. Instead of implicit learning with a surrogate objective [29, 31, 42, 50], we generate self-supervised pairs of image regions by random scaling and rotating and then train a model to predict pose value distributions so that their relative difference is consistent with scaling and rotating being used. In contrast to the previous learning-based methods, we advocate the histogram output for pose, which is similar to SIFT [22], and propose a histogram alignment technique for self-supervised learning. The method learns a non-parametric and multi-modal distributions of scale and orientation without any human annotations, effectively resolving the challenge of defining and annotating characteristic poses for image regions. Experimental results show a significant improvement over the previous method both in scale and orientation estimation on the proposed PatchPose dataset and the HPatches [1] dataset, demonstrating the effectiveness of our self-supervised learning framework. Moreover, the image matching result on HPatches [1] shows the patch extraction effect to mean matching accuracy (MMA) using our method. The 6 DoF pose estimation results on IMC2021 [17] show the outlier rejection effect by our scale and orientation. The code and models are publicly available at [this link].

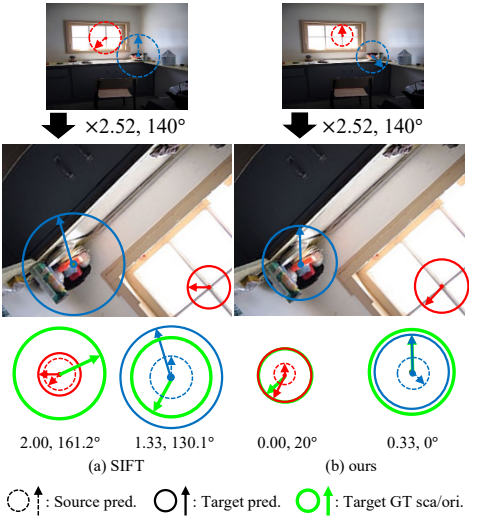


Figure 1: Comparison of scale/orientation estimation of SIFT [22] vs. ours. The size of circles represents the scale, and the direction of arrows means the orientation. At the bottom, the green circle/arrow depicts the true scale/orientation given the estimation of the top and the numbers mean the errors in relative scale/orientation.

2 Related work

Scale and orientation estimation. The most representative is the scale-invariant feature transform (SIFT) [22], where Lowe introduces gradient histograms for orientation estimation and difference of Gaussians for scale estimation. Despite its success, it often fails when geometric or photometric deformation is present. Bay *et al*. [4] improve SIFT by Hessian-based descriptors and integral images. Rublee *et al*. [38] propose efficient measure of corner orientation using intensity centroid [36] on the FAST detectors [37]. These classical methods use handcrafted algorithms to obtain scale and orientation without learning. Recent research has investigated learning-based methods to estimate characteristic scale and/or orientation for image patches. Yi *et al*. [50] introduce a CNN that learns to predict the characteristic orientation of an image patch. To avoid the difficulty of defining the characteristic orientation, they train the CNN by minimizing the distance between orientation-normalized descriptors

of two matchable patches. In the subsequent work of [31, 42], they integrate scale/orientation estimation with feature point detection and description for image matching. More recent studies [2, 10, 20, 32] aim to extract local descriptors that are invariant or covariant with respect to geometric variations within a local region. The aforementioned learning-based methods all share common strategies: (1) regression-based estimation, (2) implicit learning by improving descriptor matching, and (3) the use of matchable pairs obtained from different datasets, e.g. phototourism [26, 45, 48], ScanNet [7] with depth information and HPatches [1] with ground-truth homography. In contrast, our method uses (1) histogram-based estimation, (2) self-supervised explicit learning, and (3) unsupervised datasets with random transformation.

There also exists previous work on estimating more general transformation beyond scale and orientation. For example, Mikolajczyk and Schmid [25] introduce a scale/affine-invariant keypoints detector using an affine shape estimator based on the second moment matrix. Mishkin *et al*. [29] propose to learn an affine-covariant region detector using the spatial transformer network [16] and the triplet margin loss. In most cases of current image matching applications, however, the use of scale and orientation only is still dominant.

Self-supervised learning. The task of orientation prediction has often been used as a pretext task for self-supervised representation learning. Gidaris *et al*. [13] introduce a classification task of predicting a rotated angle of an image for representation learning. Feng *et al*. [12] propose to decouple the rotation discrimination from instance discrimination. While learning to estimate image orientation in a self-supervised manner, the predicted orientation from these methods cannot be used for the characteristic orientation we consider in this work; they assume a fixed and predefined canonical orientation (*i.e*., upright) for each object class and simply predict the rotation from it by learning the class information, which cannot generalize to arbitrary images to be aligned. In contrast, our method does not assume any prior information about predefined object classes and their canonical orientations.

Invariant feature learning. Our approach to estimating characteristic scale and orientation is also relevant to learning image relations for invariant feature representation [23, 24, 44]. Memisevic and Hinton [24] approximate a three-dimensional interaction tensor of a higher-order Boltzmann machine via factorizing the tensor. They investigate how image transformations affect the filters of the proposed model in a visual analogy task. Memisevic [23] proposes a conservative detector, called the subspace rotation detector, which generates a content-independent representation. Sohn and Lee [44] extend the RBM to capture transformation between eigenfeatures of two images. With the predicted transformation, their model extracts transformation-invariant features, which are beneficial in image classification.

The contribution of this paper is three-fold. First, we introduce a self-supervised learning framework to estimate the characteristic scale and orientation for an arbitrary image patch. Second, we propose a histogram alignment technique for learning to estimate multi-modal distribution of scale/orientation. Third, experimental evaluation on scale/orientation estimation benchmarks demonstrates the effectiveness of our approach, significantly outperforming recent methods.

3 Method

In this section, we introduce the patch pose estimation network that learns to predict characteristic scale and orientation of an image patch. We first describe the model architecture and then explain our strategy for self-supervised learning.

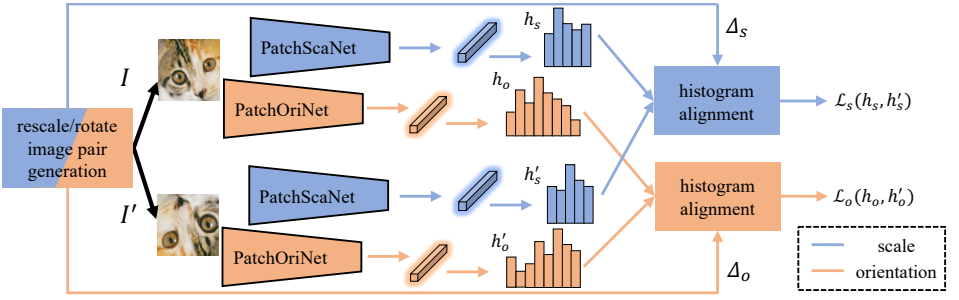


Figure 2: Overview of our self-supervised framework for learning a patch pose, *i.e.*, orientation and scale. Given a pair of image patches with rescaling/rotating, we feed them to the patch pose estimation networks that output scale/orientation histograms for each image patch. We compare the two histograms by the histogram alignment technique and compute the loss, which is used for training the networks via backpropagation.

3.1 Patch pose estimation networks

The patch pose estimation networks are designed to predict the characteristic pose, *i.e.*, orientation and scale, of a given image patch. We cast the patch pose estimation into the problem of predicting a probability distribution over candidate pose values rather than that of regressing a target pose value. The basic form of the architecture thus consists of a convolutional feature extractor followed by MLPs with softmax output that produces a histogram over a set of candidate pose values:

$$h = \sigma(\texttt{MLP}(\texttt{CONV}(I))) \tag{1}$$

where $\sigma(\cdot)$ is the softmax function and $h \in \{x \in \mathbb{R} : 0 \leq x \leq 1\}^B$ is the histogram distribution of pose with B bins, *i.e.*, either orientation or scale. Figure 2 shows the overall architecture of our model.

The output histogram h, *i.e.*, B bins of discretized candidate pose values for either orientation or scale, represents a distribution over the pose values. In contrast to previous regression-based methods [29, 31, 42, 50], which predict only a single pose, our histogram estimator is able to naturally predict multiple plausible poses by a multi-modal histogram distribution, and can be effectively trained with our self-supervised objective. For scale estimation, inspired by the scale space of SIFT [22], we create 13 bins over the $\log_2$ scale space, *i.e.*, $B_s = 13$, which are centered on $\{-2, -\frac{5}{3}, ..., 0, ..., \frac{5}{3}, 2\}$ so that each bin covers the span of $\frac{4}{B_s - 1}$ in $\log_2$ scale from its center. For orientation estimation, we create 36 bins over 2π, *i.e.*, $B_o = 36$, which

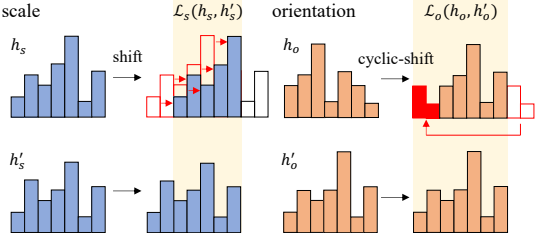


Figure 3: Histogram alignment for scale and orientation. The histogram h is shifted by a ground-truth Δ. For scale, the overlapping regions of the shifted one and the other are used to compute the loss. For orientation, the shift operation is circular so that the entire regions of histograms are used.

are centered on $\{0, \frac{\pi}{18}, ..., \frac{35\pi}{18}\}$ so that each bin covers the span of $\frac{2\pi}{B_o}$ in radians from its center.

We train our model using a self-annotated dataset of image patch pairs that are generated by transforming images with random rescaling/rotating. Let us assume such a dataset of image patches $\mathcal{D} = \{(I_n, I'_n, \Delta_n)\}_{n=1}^{N}$, where Δ_n denotes the ground-truth relative pose from I_n to I'_n. Note that we do not have a manually labeled pose for either of the two image patches; the relative pose difference between them is the only supervisory signal we exploit. To train our model using the limited self-supervision, we propose the *histogram alignment* loss that aligns one histogram to the other and then measures the discrepancy between the aligned histograms. In training, a pair of image patches, I and I', from the dataset $\mathcal{D}$ are fed into our model to predict pose histograms of the two patches, h and h', respectively. Figure 3 illustrates the concept of the histogram alignment losses, which will be detailed subsequently.

Scale. We define the histogram shift operator T^d ($d \in \mathbb{R}$) that takes a histogram h on $\mathbb{Z}$ and translates it to the left by d with a linear interpolation:

$$T^d h(i) = \begin{cases} h\big(i+d\big) & \text{if } d \in \mathbb{Z} \\ \big(\lceil d \rceil - d\big) h\big(i + \lfloor d \rfloor\big) + \big(d - \lfloor d \rfloor\big) h\big(i + \lceil d \rceil\big) & \text{otherwise,} \end{cases} \tag{2}$$

where $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$ denotes the floor and the ceiling, respectively, and any other interpolation can replace the linear one. This enables the shifting operator T^d to cover a non-integer number d in general.

Let us consider an image I and its scaled image I' by Δ_s in $\log_2$ scale. To align their scale histogram outputs, h_s and h'_s, we shift h'_s by $\frac{(B_s-1)\Delta_s}{4}$, since $\frac{4}{B_s-1}$ is a single bin coverage per $\log_2$ scale. Given the bins of h_s, indexed by $\{0, 1, ..., B_s - 1\}$, the set of bins $\mathcal{B}$ that shares the same scales with the shifted scale histogram $T^{\frac{(B_s-1)\Delta_s}{4}} h'_s$ is

$$\mathcal{B} = \begin{cases} \{i \mid 0 \le i \le B_s - 1 - \lceil \frac{(B_s-1)\Delta_s}{4} \rfloor)\} & \text{if } \Delta_s \ge 0 \\ \{i \mid -\lceil \frac{(B_s-1)\Delta_s}{4} \rfloor \le i \le B_s - 1\} & \text{if } \Delta_s < 0, \end{cases} \tag{3}$$

where $\lceil \cdot \rfloor$ denotes the rounding to the nearest integer.

Finally, given the ground-truth scale shift Δ_s from I to I', the histogram alignment loss for scale computes the distance between the shared parts of the two scale histograms aligned by the histogram shift:

$$\mathcal{L}_s(h_s, h'_s) = -\sum_{i \in \mathcal{B}} h_s(i) \log\big(T^{\frac{(B_s-1)\Delta_s}{4}} h'_s(i)\big), \tag{4}$$

where only the bins of shared scales contribute to the loss. We use the cross-entropy to enforce the two histograms to match.

Orientation. To handle the circular property of orientation, we define the circular shift operator T_B^d on $\{0, 1, ..., B-1\}$:

$$T_B^d h(i) = \begin{cases} h\big((i+d) \bmod B\big) & \text{if } d \in \mathbb{Z} \\ \big(\lceil d \rceil - d\big) h\big((i + \lfloor d \rfloor) \bmod B\big) + \big(d - \lfloor d \rfloor\big) h\big((i + \lceil d \rceil) \bmod B\big) & \text{otherwise,} \end{cases} \tag{5}$$

where the modulo operation uses the floored division so that the output is a non-negative integer. Note that this histogram shift can cover any rotation value of $d \in \mathbb{R}$.

Let us consider an image I and its rotated image I' by Δ_o in radians. To match their orientation histogram outputs, h_o and h'_o, we circular-shift h'_o by $\frac{B_o \Delta_o}{2\pi}$, since $\frac{2\pi}{B_o}$ is a single bin coverage per radian. As the result, the bins of each histogram, indexed by $\{0, 1, ..., B_o - 1\}$, are aligned to have the same orientation. Therefore, given the ground-truth orientation shift Δ_o from I to I', the histogram alignment loss for orientation computes the distance between the two orientation histograms aligned by the circular shift:

$$\mathcal{L}_o(h_o, h'_o) = -\sum_{i=0}^{B_o-1} h_o(i) \log \left(T_{B_o}^{\frac{B_o \Delta_o}{2\pi}} h'_o(i) \right). \tag{6}$$

These two losses allow us to train the scale and orientation estimators without characteristic scale and orientation annotations, by defining the characteristic scale and orientation of an image patch in a relative manner which are consistently estimated with the other corresponding patch. The overall training objectives for scale and orientation estimation are

$$\mathcal{L}_s = \mathcal{L}_s(h_s, h'_s) + \mathcal{L}_s(h'_s, h_s), \quad \mathcal{L}_o = \mathcal{L}_o(h_o, h'_o) + \mathcal{L}_o(h'_o, h_o), \tag{7}$$

where we use an additional term to make the objectives symmetric for the two histograms.

4 Experiments

We conduct experiments to demonstrate the efficacy of our method. In this section, we explain the implementation details, describe datasets with their evaluation metrics, and then show experimental results with in-depth analyses.

4.1 Implementation details

We use the ResNet-18 [15] backbone as a feature extractor and train the whole networks from random initialization. We use two separate models for scale and orientation estimators. Both the estimators are implemented with four-layer MLPs. We resize the input patch size $I \in \mathbb{R}^{3\times32\times32}$. Our model yields an orientation vector $h_o \in \mathbb{R}^{36}$ and a scale vector $h_s \in \mathbb{R}^{13}$ as outputs. We use a batch size 64, a SGD optimizer with a learning rate 3.0 and a momentum 0.9. We set the softmax temperature value to 20 for stable learning.

Inference. We use a simple arg max function to convert the histogram to a single pose value.

$$f_s(I) = 2^{\frac{4}{B_s - 1} \arg\max_i (h_s(i))}, \quad f_o(I) = \frac{2\pi}{B_o} \arg\max_i (h_o(i)), \tag{8}$$

where B_s and B_o are the numbers of bins for scale and orientation, respectively. Figure 4 visualizes predicted orientation/scale histograms for different input patches. While further non-maximum suppression or smoothing schemes can also be adopted for the histograms [19], we use the simple arg max function to determine the final pose in our work.

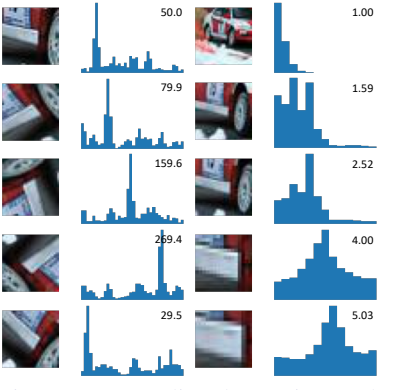


Figure 4: Predicted rotation/scale histograms for different inputs. The numbers denote the estimated pose values. For better visualization, we suppress non-maximal values in the histograms via softmax.

4.2 Evaluation Benchmarks

We use three datasets, PatchPose, HPatches [1] and IMC2021 [17]. The PatchPose dataset is constructed by us for learning and evaluation; our model is trained on its train split and tested on its test split. HPatches [1] and IMC2021 [17] are employed to evaluate the transferability of the learned model; they are used for evaluation only.

PatchPose dataset generation. The PatchPose dataset is synthetically generated from 1,793 images of SPair-71k [27] from PASCAL-VOC [11]. We extract 3 keypoints of an image using SIFT [22], on which 64×64 patches are centered on to be cropped after transformed by $\Delta_s, \Delta_o \in \mathbb{R}$. We pair the source patch and its augmented patch for inputs to the network. The dataset of patch pairs with relative pose annotation is generated without manual annotation. The used rescaling and rotating degrees, Δ_s and Δ_o, are annotated for free. The values for rescaling Δ_s are distributed in the range of $[2^{-2}, 2^2]$ and those for rotating Δ_o are in the range of $[0, 2\pi)$, covering wide ranges of scale and orientation changes. The dataset is split as train : val : test = 3,947,054 : 40,276 : 40,278. For details, see the supplementary material.

Evaluation metric. To evaluate predicted poses of two image patches, we define accuracy metrics. We first measure the errors using $\log_2$-scale and radian-orientation differences:

$$s(I, I'; f_s, \Delta_s) = |\log_2(\frac{f_s(I')}{f_s(I)}) - \Delta_s|, \;\; o(I, I'; f_o, \Delta_o) = |(f_o(I') - f_o(I)) \bmod 2\pi - \Delta_o|, \quad (9)$$

where I and I' are the image pair with known difference in scale Δ_s and that in orientation Δ_o. f_s and f_o are scale and orientation estimators, respectively. We then convert the errors to accuracy values using some thresholds, *i.e.*, $\{\frac{1}{6}, \frac{1}{3}\}$ for scale and $\{\frac{\pi}{36}, \frac{\pi}{18}\}$ for orientation.

Transferability evaluation. We also use the HPatches [1] viewpoint variation for transferability evaluation. The HPatches viewpoint variation has 59 scenes; each scene has 6 images with known homography matrices. The ground-truth scale and orientation are extracted from homography $A_{3\times3}$ by

$$\Delta_s = \sqrt{(\frac{A_{11}}{A_{33}})^2 + (\frac{A_{21}}{A_{33}})^2}, \quad \Delta_o = \arctan(\frac{A_{21}}{A_{11}}). \quad (10)$$

We extract 25 patches centered on SIFT [22] and Harris [14] keypoints, and then extract patches centered on the corresponding keypoints from the other image. As the result, we sample 7,375 patch pairs used for the pose estimation evaluation on HPatches [1]. On the other hand, we use the all 116 sequences (59 viewpoint, 57 illumination) of HPatches [1] to evaluate our method on image matching. HPatches is an image matching benchmark with ground-truth homography. We evaluate our patch extraction ability on image matching pipeline compared to the existing methods [22, 28, 47]. For each sequence, we pair the first image to 5 other images, so a total of 580 image pairs are used. To evaluate patch extraction on the image matching, we use the number of matches and mean matching accuracy (MMA) as evaluation metrics.

To demonstrate the effectiveness of our method on a more complex dataset, we use the IMC2021 [17] wide-baseline matching benchmark. IMC2021 [17] consists of an unconstrained urban scene with large illumination and viewpoint variations; the validation set of Phototourism and Prague parks are used to evaluate our method. This benchmark takes matches as input and measures the quality of 6 DoF pose estimation. We use our predicted patch scale/orientation for the outlier rejection [5] scheme in the image matching pipeline [2, 5, 6, 22, 28] and measure the mean average accuracy (mAA) at 5°and 10°of the pose estimation and the number of inliers as evaluation metrics.

[1] We measure the scores of OriNet [30], AffNet [29], LF-Net [31] and RF-Net [32] using official released code by authors. SIFT [22] score is measured by modification of OpenCV model.

methods	PatchPose				HPatches			
	sca. ($\log_2$)		ori. (radian)		sca. ($\log_2$)		ori. (radian)	
	$\pm\frac{1}{6}$	$\pm\frac{1}{3}$	$\pm\frac{\pi}{36}$	$\pm\frac{\pi}{18}$	$\pm\frac{1}{6}$	$\pm\frac{1}{3}$	$\pm\frac{\pi}{36}$	$\pm\frac{\pi}{18}$
SIFT [22]	28.3	44.9	15.5	28.7	11.3	25.3	11.2	25.6
OriNet [50]	-	-	29.1	45.0	-	-	15.8	29.8
LF-Net [31]	10.6	17.6	13.7	25.3	8.1	25.5	14.2	24.5
AffNet [29]	-	-	27.0	42.0	-	-	14.0	23.9
RF-Net [42]	10.6	17.4	4.0	6.6	7.8	26.1	15.6	32.9
ours	**57.9**	**78.2**	**80.5**	**97.9**	**29.0**	**53.0**	**52.0**	**69.2**

Table 1: Accuracy of patch pose estimation on the PatchPose and the HPatches viewpoint variation. The bold numbers indicate the best and the underlined ones are the second best.[1]

4.3 Patch Pose Estimation

We evaluate our method, which is trained using the PatchPose training split, and compare it with the other methods [22, 29, 31, 42, 50] on the PatchPose test split and the HPatches [1] viewpoint variation.

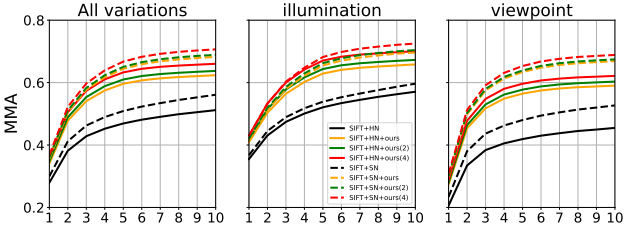
PatchPose. The left side of Table 1 shows patch pose estimation results on the PatchPose test split compared to the existing methods [22, 29, 31, 42, 50]. Our method outperforms the previous methods by a large margin at all thresholds in both scale/orientation estimation. In particular, our orientation estimator achieves an almost perfect accuracy of 97.9% at $\frac{\pi}{18}$ threshold. The regression-based learning methods [29, 31, 42, 50], which learn to estimate scale/orientation implicitly by improving descriptor similarity, turn out to perform significantly worse than ours.

HPatches. The right side of Table 1 shows the results on the HPatches [1] viewpoint variation. The orientation estimation results of RF-Net [42] on HPatches [1] performs better than those on PatchPose; we find this is because (1) RF-Net is trained on the subset of HPatches and (2) the limited range of RF-Net orientation prediction coincides more with the true orientation range of HPatches. Note that the other methods [29, 31, 50], including ours, have not been trained on this HPatches dataset. These results show that our self-supervised model transfers well to unseen patches from a different domain with unseen transformations. For all the methods, the scores on HPatches are lower than those on PatchPose due to the shear and/or tilt factors of transformation in image pairs from HPatches, which renders scale/orientation prediction more challenging.

Multi-pose estimation. Unlike regression-based methods [29, 31, 42, 50], the histogram-based methods, ours and SIFT [22], can naturally leverage multiple candidates of scale and orientation for each image patch by selecting multiple modes from the predicted histograms. To observe the potential gain of using multiple candidates, we select the top-k scale/orientation candidates for each image patch and measure whether a true pair of scale/orientation predictions is present between two corresponding sets of the top-k candidates. Table 2 shows the recall performance on the PatchPose test split, where we vary the number of candidates from 1 to 4.

	top-k	sca. ($\log_2$)		ori. (radian)	
		$\pm\frac{1}{6}$	$\pm\frac{1}{3}$	$\pm\frac{\pi}{36}$	$\pm\frac{\pi}{18}$
SIFT	top-1	29.3	44.9	15.5	28.7
	top-2	55.4	65.2	29.4	44.3
	top-3	68.6	74.8	41.6	57.1
	top-4	78.0	84.7	59.3	75.6
ours	top-1	57.9	78.2	80.5	97.9
	top-2	76.4	84.8	99.0	99.4
	top-3	83.4	88.8	99.8	99.9
	top-4	**87.8**	**93.4**	**99.9**	**100.0**

Table 2: Recall of histogram-based methods on PatchPose using top-k candidates.



methods		M	MMA
Det. + Des.	Pose.		3px
SIFT+HN [22, 28]	SIFT	148.9	0.45
	ours	154.6	0.57
	ours (top2)	153.8	0.59
	ours (top4)	**157.6**	**0.61**
SIFT+SN [22, 47]	SIFT	144.3	0.49
	ours	158.8	0.62
	ours (top2)	157.7	0.62
	ours (top4)	**160.1**	**0.64**

Figure 5: Mean matching accuracy (%) with off-the-shelf keypoint detectors and descriptors on HPatches. The number beside ours means the number of top-k candidates. 'M' denotes the average number of matches. We fix the average number of keypoints all the same.

The results show that our method substantially outperforms the classical histogram-based method, SIFT [22]. In particular, our model achieves 100% recall of orientation estimation in top-4 selection at $\frac{\pi}{18}$ threshold. The use of multiple candidates allows effective image matching in Sec. 4.4, which is not available for regression-based methods.

Effect of histogram sizes. Table 3 shows the accuracy variations with different numbers of bins B_s and B_o. As the number increases to a proper value, the histogram becomes more fined-grained and thus the prediction tends to be more precise. However, when it increases further (*e.g.*, $B_s = 17, B_o = 72$), we find the training process becomes unstable due to the increased classes for prediction. We thus set the values as $B_s = 13$ and $B_o = 36$.

B_s	sca. ($\log_2$)		B_o	ori. (radian)	
	$\pm\frac{1}{6}$	$\pm\frac{1}{3}$		$\pm\frac{\pi}{36}$	$\pm\frac{\pi}{18}$
7	22.5	22.5	9	26.5	26.5
9	37.3	37.3	18	48.7	48.7
13	**57.9**	**78.2**	36	**80.5**	**97.9**
17	19.7	35.6	72	9.7	15.4

Table 3: Accuracy of patch pose estimation on PatchPose with different numbers of bins B.

4.4 Application to Image Matching

We validate our patch pose estimators by applying them to image matching. In the matching pipeline, keypoints and their scale/orientation pose values are extracted from images by an existing detector. Basically, we replace their pose values with our results for comparison.

Evaluation on HPatches. In this matching accuracy evaluation, two sets of image patches are extracted from an image pair using detected keypoints and their estimated patch poses[2] and then are matched via mutual nearest neighbors according to the similarity of patch descriptors; SIFT [22] is used for keypoint detection while HardNet [28] and SOSNet [47] are for patch description. In this matching pipleline, we use our pose estimation for image patch extraction and evaluate its effect. To leverage our multi-pose estimation in matching, we extract multiple patches for each keypoint using its top-k poses. Figure 5 shows the image matching results on HPatches [1], where the use of our method for patch extraction consistently improves over the baseline methods. Even without multi-pose estimation, our method achieves better Mean Matching Accuracy (MMA) than all of the baselines. Our result with the top-4 pose estimation improves both MMA and the number of matches. It shows that our method transfers well to image matching without any fine-tuning on the target datasets.

Evaluation on IMC2021. In this 6 DoF camera pose estimation evaluation, we collect

[2]To avoid sampling patches from outside of the image, we exclude keypoints near boundaries, *i.e.* $(w < 16) \vee (h < 16) \vee (w > W - 16) \vee (h > H - 16)$ where (W, H) denotes the image size and (w, h) is the keypoint coordinate.

Det.+Pose.	K	Phototourism			Prague parks		
		Num. Inl.	mAA(5°)	mAA(10°)	Num. Inl.	mAA(5°)	mAA(10°)
SIFT+AffNet [22, 29]	1,024	46.4	0.250	0.321	35.9	0.090	0.145
SIFT+ours	1,024	**60.8**	**0.316**	**0.397**	**51.3**	**0.197**	**0.277**
SIFT+AffNet [22, 29]	2,048	110.9	0.448	0.542	90.7	0.196	0.282
SIFT+ours	2,048	**131.7**	**0.471**	**0.566**	**112.8**	**0.291**	**0.401**
Key.Net [2]	1,024	75.4	0.319	0.409	**175.9**	0.422	0.562
Key.Net+ours	1,024	**77.6**	**0.329**	**0.420**	175.4	**0.443**	**0.583**
Key.Net [2]	2,048	167.5	0.431	0.537	**368.8**	0.514	**0.660**
Key.Net+ours	2,048	**172.4**	**0.446**	**0.553**	368.7	**0.518**	**0.660**

Table 4: Mean average accuracy (mAA; 5°, 10°) of 6-DoF pose estimation and the number of inlier matches (Num. Inl.) on IMC2021 [17] validation set.

reliable feature matches and use them to estimate a camera pose[3] via the standard structure-from-motion method [40]. To obtain the set of reliable matches, we first obtain mutual nearest neighbor matches via a standard feature extraction and matching process [22, 28, 29], and then purify those matches using the outlier rejection method of AdaLAM [5] and the robust model fitting of DEGENSAC [6]. In this pipeline, we use our estimated patch poses for the outlier rejection step of AdaLAM [5]. For comparison, we use SIFT+AffNet [22, 29] and Key.Net [2] as two baselines for keypoint detection and patch pose estimation, and evaluate the effect of replacing their orientation estimation with ours in the process of outlier rejection. For all the methods, HardNet [28] is used for patch description. Table 4 shows the results of 6 DoF pose estimation in the validation set of the IMC2021 stereo task [17]. Our method improves over SIFT+AffNet [22, 29] and Key.Net [2] in 6 DoF pose estimation accuracy on both Phototourism and Prague parks. The performance gain of our method on Key.Net is smaller than that on SIFT+AffNet [22, 29]. We find this is due to the keypoint selection scheme of Key.Net [2]; it selects the keypoints based on the local window so that they spread evenly, which reduces the impact of the subsequent outlier rejection step.

5 Conclusion

We have proposed a self-supervised learning framework for characteristic scale and orientation estimation. Our method effectively estimates characteristic scale and orientation via the histogram alignment technique. Our experiments show impressive results on PatchPose and HPatches datasets, achieving the state-of-the-art performance on the task of scale and orientation estimation. Moreover, the use of our patch pose estimation has been shown to improve matching performance on HPatches and IMC2021 benchmarks, on which our method has never been trained. We believe further research in this direction can benefit a variety of image matching, visual localization, and recognition problems in computer vision.

Acknowledgement

This work was supported by Samsung Electronics Co., Ltd., the NRF grant (NRF-2021R1A2C3012728; NRF-2019H1A2A1076171 - Global Ph.D. Fellowship Program), and the IITP grant (No.2019-0-01906, AI Graduate School Program - POSTECH) funded by Ministry of Science and ICT, Korea.

[3] We evaluate to use the provided source code from IMC2021. https://github.com/ubc-vision/image-matching-benchmark

References

[1] Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5173–5182, 2017.

[2] Axel Barroso-Laguna, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. Key. net: Keypoint detection by handcrafted and learned cnn filters. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5836–5844, 2019.

[3] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.

[4] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. In *European conference on computer vision*, pages 778–792. Springer, 2010.

[5] Luca Cavalli, Viktor Larsson, Martin Ralf Oswald, Torsten Sattler, and Marc Pollefeys. Handcrafted outlier detection revisited. In *European Conference on Computer Vision*, pages 770–787. Springer, 2020.

[6] Ondrej Chum, Tomas Werner, and Jiri Matas. Two-view geometry estimation unaffected by a dominant plane. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 772–779. IEEE, 2005.

[7] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5828–5839, 2017.

[8] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 224–236, 2018.

[9] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-net: A trainable cnn for joint detection and description of local features. *arXiv preprint arXiv:1905.03561*, 2019.

[10] Patrick Ebel, Anastasiia Mishchuk, Kwang Moo Yi, Pascal Fua, and Eduard Trulls. Beyond cartesian representations for local descriptors. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 253–262, 2019.

[11] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[12] Zeyu Feng, Chang Xu, and Dacheng Tao. Self-supervised representation learning by rotation feature decoupling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10364–10374, 2019.

[13] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

[14] Chris Harris and Mike Stephens. A combined corner and edge detector. In *In Proc. of Fourth Alvey Vision Conference*, pages 147–151, 1988.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[16] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015.

[17] Yuhe Jin, Dmytro Mishkin, Anastasiia Mishchuk, Jiri Matas, Pascal Fua, Kwang Moo Yi, and Eduard Trulls. Image matching across wide baselines: From paper to practice. *International Journal of Computer Vision*, 129(2):517–547, 2021.

[18] Jongmin Lee, Yoonwoo Jeong, Seungwook Kim, Juhong Min, and Minsu Cho. Learning to distill convolutional features into compact local descriptors. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 898–908, 2021.

[19] Junghyup Lee, Dohyung Kim, Jean Ponce, and Bumsub Ham. Sfnet: Learning object-aware semantic correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2278–2287, 2019.

[20] Yuan Liu, Zehong Shen, Zhixuan Lin, Sida Peng, Hujun Bao, and Xiaowei Zhou. Gift: Learning transformation-invariant dense visual descriptors via group cnns. *arXiv preprint arXiv:1911.05932*, 2019.

[21] Jonathan L Long, Ning Zhang, and Trevor Darrell. Do convnets learn correspondence? In *Advances in neural information processing systems*, pages 1601–1609, 2014.

[22] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[23] Roland Memisevic. On multi-view feature learning. In *ICML*, 2012.

[24] Roland Memisevic and Geoffrey E Hinton. Learning to represent spatial transformations with factored higher-order boltzmann machines. *Neural computation*, 22(6): 1473–1492, 2010.

[25] Krystian Mikolajczyk and Cordelia Schmid. Scale & affine invariant interest point detectors. *International journal of computer vision*, 60(1):63–86, 2004.

[26] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE transactions on pattern analysis and machine intelligence*, 27(10): 1615–1630, 2005.

[27] Juhong Min, Jongmin Lee, Jean Ponce, and Minsu Cho. Spair-71k: A large-scale benchmark for semantic correspondence. *arXiv preprint arXiv:1908.10543*, 2019.

[28] Anastasiia Mishchuk, Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. In *Advances in Neural Information Processing Systems*, pages 4826–4837, 2017.

[29] Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Repeatability is not enough: Learning affine regions via discriminability. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 284–300, 2018.

[30] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-scale image retrieval with attentive deep local features. In *Proceedings of the IEEE international conference on computer vision*, pages 3456–3465, 2017.

[31] Yuki Ono, Eduard Trulls, Pascal Fua, and Kwang Moo Yi. Lf-net: learning local features from images. In *Advances in neural information processing systems*, pages 6234–6244, 2018.

[32] Rémi Pautrat, Viktor Larsson, Martin R Oswald, and Marc Pollefeys. Online invariance selection for local feature descriptors. In *European Conference on Computer Vision*, pages 707–724. Springer, 2020.

[33] Filip Radenović, Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondřej Chum. Revisiting oxford and paris: Large-scale image retrieval benchmarking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5706–5715, 2018.

[34] Jerome Revaud, Philippe Weinzaepfel, César De Souza, Noe Pion, Gabriela Csurka, Yohann Cabon, and Martin Humenberger. R2d2: Repeatable and reliable detector and descriptor. *arXiv preprint arXiv:1906.06195*, 2019.

[35] Ignacio Rocco, Mircea Cimpoi, Relja Arandjelovic, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Ncnet: Neighbourhood consensus networks for estimating image correspondences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[36] Paul L Rosin. Measuring corner properties. *Computer Vision and Image Understanding*, 73(2):291–307, 1999.

[37] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *European conference on computer vision*, pages 430–443. Springer, 2006.

[38] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pages 2564–2571. Ieee, 2011.

[39] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, et al. Benchmarking 6dof outdoor visual localization in changing conditions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8601–8610, 2018.

[40] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016.

[41] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016.

[42] Xuelun Shen, Cheng Wang, Xin Li, Zenglei Yu, Jonathan Li, Chenglu Wen, Ming Cheng, and Zijian He. Rf-net: An end-to-end image matching network based on receptive field. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8132–8140, 2019.

[43] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[44] Kihyuk Sohn and Honglak Lee. Learning invariant representations with local transformations. In *ICML*, 2012.

[45] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016.

[46] Yurun Tian, Bin Fan, and Fuchao Wu. L2-net: Deep learning of discriminative patch descriptor in euclidean space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 661–669, 2017.

[47] Yurun Tian, Xin Yu, Bin Fan, Fuchao Wu, Huub Heijnen, and Vassileios Balntas. Sosnet: Second order similarity regularization for local descriptor learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11016–11025, 2019.

[48] Kyle Wilson and Noah Snavely. Robust global translations with 1dsfm. In *European Conference on Computer Vision*, pages 61–75. Springer, 2014.

[49] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. Lift: Learned invariant feature transform. In *European Conference on Computer Vision*, pages 467–483. Springer, 2016.

[50] 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*, pages 107–116, 2016.