ROTATION-EQUIVARIANT KEYPOINT DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

We show how to train a rotation-equivariant representation to extract local keypoints for image matching. Existing learning-based methods focused on extracting translation-equivariant keypoints using conventional convolutional neural networks (CNNs), but rotation-equivariant keypoint detectors have not been studied extensively. Therefore, we propose a rotation-invariant keypoint detection method using rotation-equivariant CNNs. Our rotation-equivariant representation enables us to estimate local orientations to image keypoints accurately. We propose a dense histogram alignment loss to assign an orientation to keypoints more consistently. We validate the effectiveness compared to existing keypoint detection methods. Furthermore, we check the transferability of our method on public image matching benchmarks.

1 INTRODUCTION

Extracting keypoints robust to imaging variations is crucial for computer vision problems such as image matching, structure-from-motion (SfM), and 3D reconstruction. Conventional image matching pipeline obtains a correspondence set by obtaining keypoints and extracting descriptors, then finding the similarity of descriptors corresponding to those keypoints. In the deep learning era, several dense matching methods (DeTone et al., 2018; Dusmanu et al., 2019; Rocco et al., 2017; 2018a;b; Seo et al., 2018; Choy et al., 2016; Min et al., 2019; Lee et al., 2021; Noh et al., 2017; Truong et al., 2020b;a; Kim et al., 2018; Jiang et al., 2021; Sun et al., 2021) find correspondences by correlation tensor with a dense descriptor without a separate keypoint detector on the input image. On the other hand, sparse matching methods find correspondences to extract sparse keypoints integrating with patch-based descriptor extraction methods (Lowe, 2004; Mikolajczyk & Schmid, 2004; Bay et al., 2006; Rublee et al., 2011; Revaud et al., 2019; Ono et al., 2018; Shen et al., 2019; Barroso-Laguna et al., 2019; Novotny et al., 2017; Mishchuk et al., 2017; Tian et al., 2017; 2019; 2020). The existing dense matching methods propose simple end-to-end image matching networks, but the dense matching methods without accurate keypoint detection fail to find accurate matches on viewpoint change. Therefore, keypoint detection that is accurate and consistent with geometric changes is still an important problem in image matching.

Representative keypoint detection method (Lowe, 2004) extracts difference-of-Gaussian (DoG) feature on scale-space to consider scale-invariant and then find the local extrema from the DoG feature. Recently, keypoint detection research is conducted together with descriptor extraction in an end-to-end manner (Ono et al., 2018; Shen et al., 2019) or train the keypoint extraction separately (Barroso-Laguna et al., 2019). Even though most of them obtain scale-equivariant features using scale-space to obtain scale-invariant in local features, but rotation-equivariant representations are not properly considered in the keypoint detection task. Therefore, we propose a method to extract rotation-invariant keypoints using a rotation-equivariant representation through specially designed rotation-equivariant convolutional layers (Cohen et al., 2019). The rotation-equivariant representation has the advantages of explicitly encoding the enriched orientation information and reducing the model size through weight sharing compared to the regular representation.

The rotation-equivariant representation utilizes to assign orientations to keypoints. Existing local orientation assignment to keypoints uses the form of a histogram that aggregates image gradients. Recently, existing learning methods to assign local orientations train the orientation by implicit way using descriptor similarity in the image matching pipeline, and there is no research to learn local orientation by giving explicit loss, in our knowledge. Therefore, we propose an orientation alignment loss function to estimate a characteristic orientation to the keypoints using explicit supervision. This

is done in a self-supervised manner using synthetically generated pairs augmented through random rotation.

To show the effectiveness of our model, we compare it with handcrafted model (Lowe, 2004) and learning-based model (Barroso-Laguna et al., 2019) on image matching benchmarks. We evaluate with patch-based descriptors (Mishchuk et al., 2017; Tian et al., 2019; 2020) using repeatability score and matching accuracy in order to verify the effectiveness of keypoint detection. Our estimated orientations improve the keypoint matching accuracy with outlier rejection on the HPatches benchmark (Balntas et al., 2017). Furthermore, we evaluate 6 DoF pose estimation benchmark, IMC2021 (Jin et al., 2021), to show the transferability in a more complex task. We conduct ablation experiment to demonstrate the effectiveness of components of our model.

The contributions of our paper are three-fold:

- We propose a self-supervised framework for learning to extract rotation-invariant keypoints using rotation-equivariant representation.
- We propose a histogram-based alignment loss to obtain orientations that change consistently to geometric transformations for the obtained keypoints.
- We show our effectiveness through several experiments by comparing the existing keypoint detection method and the extensive evaluation of the image matching benchmark.

2 Related work

This section is organized into three parts: Keypoint detection, local orientation estimation, and rotation-equivariant representation.

Keypoint detection. It plays a key role in many computer vision tasks such as image matching, SfM, and 3D reconstruction. Traditional keypoint detectors rely on carefully designed handcrafted filters. Lowe (2004) proposes keypoint detection through the local extrema in DoG, and Bay et al. (2006) boosts up the speed of keypoint detection by using Haar filters. Inspired by the recent success of deep learning to various computer vision tasks, learning-based approaches have been proposed recently to learn how to detect keypoints. Verdie et al. (2015); Yi et al. (2016a); DeTone et al. (2018); Ono et al. (2018); Shen et al. (2019) propose learning techniques of a keypoint detector through a CNNs-based response map. Barroso-Laguna et al. (2019) utilized the benefit of both handcrafted and learning approaches to improve the performance in terms of repeatability. Contrary to them, we utilize rotation-equivariant features to obtain consistent keypoint locations. Furthermore, our rotation-equivariant representation yields characteristic orientation of the keypoints, and applies it outlier rejection in an image matching pipeline.

Local orientation estimation. Lowe (2004) classically uses image gradient aggregation to estimate local orientation. Rublee et al. (2011) propose an efficient way to measure corner orientation using intensity centroid (Rosin, 1999). Learning-based methods (Yi et al., 2016b; Ono et al., 2018; Shen et al., 2019) use surrogate loss through local descriptor extraction to train local orientation in the image matching pipeline. They use orientation as one of the affine parameters in the patch sampling using STNs (Jaderberg et al., 2015). Afterward, Ebel et al. (2019) sample local patches by transforming scale and orientation parameters into a log-polar coordinate system. The aforementioned learning methods learn local orientation as implicit ways by using descriptor similarity losses. In addition, these methods predict orientation values through regression of trigonometric values, but our explicit learning method predicts orientation with histogram-based representation from the perspective of a classification problem. We utilize group CNNs for consistent orientation estimation.

Rotation-equivariant representation. An equivariant representation means the result obtained by changing the input is the same as the result by changing the output after the result is obtained. Memisevic & Hinton (2010); Memisevic (2012) propose a content-independent representation through tensor factorization based on restricted Boltzmann machine (RBM). Sohn & Lee (2012) extend RBM to capture transformation through eigenfeatures between two images. Since CNNs become popular, Cohen & Welling (2016a) propose group equivariant convolutional networks using discrete isometric groups. Marcos et al. (2017); Zhou et al. (2017) propose resampling filters using interpolation to encode explicit orientations. Weiler et al. (2018); Worrall et al. (2017) use harmonics as filters to extract equivariant features from more diverse groups and continuous domains. Weiler &

Cesa (2019) extend this group to the general $E(2)$ groups, and Sosnovik et al. (2020) propose scale-equivariant steerable networks. From an application point of view, Han et al. (2021) use rotation-equivariant networks to rotated object detection on the aerial images. Pielawski et al. (2020) apply the rotation-equivariant representation for registration of multimodal images. Likewise, we use rotation-equivariant convolution to extract repeatable keypoints consistent with various changes.

3 ROTATION-EQUIVARIANT KEYPOINT DETECTION

3.1 OVERVIEW

The goal of our work is to learn to detect oriented keypoints from images. The classical keypoint relying on handcrafted features satisfies the rotation/translation equivariance, but it is sensitive to illumination changes or color distortions. On the contrary, recent learning-based keypoint detectors use CNNs to encode local geometry and high-level semantics through convolutional layers. The convolution operation is inherently translation-equivariant, not rotation-equivariant. Therefore, We construct a rotation-equivariant model without handcrafted features to take advantage of both approaches. The proposed method consists of rotation-equivariant layers and following two branches, the keypoint detection and orientation estimation. The overall architecture is in Figure 1. The rotation-equivariant layers take an image as input and feed rotation-equivariant features into the following two branches. The keypoint detection and orientation estimation branches generate a rotation-invariant keypoint score map and a rotation-preserving orientation map through group pooling and channel pooling, respectively. Window-based keypoint detection loss and orientation alignment loss learn the keypoints and orientations, respectively, along with the ground-truth homography in a self-supervised manner. Furthermore, the multi-scale image pyramid encourages the network to have robustness on scale changes.

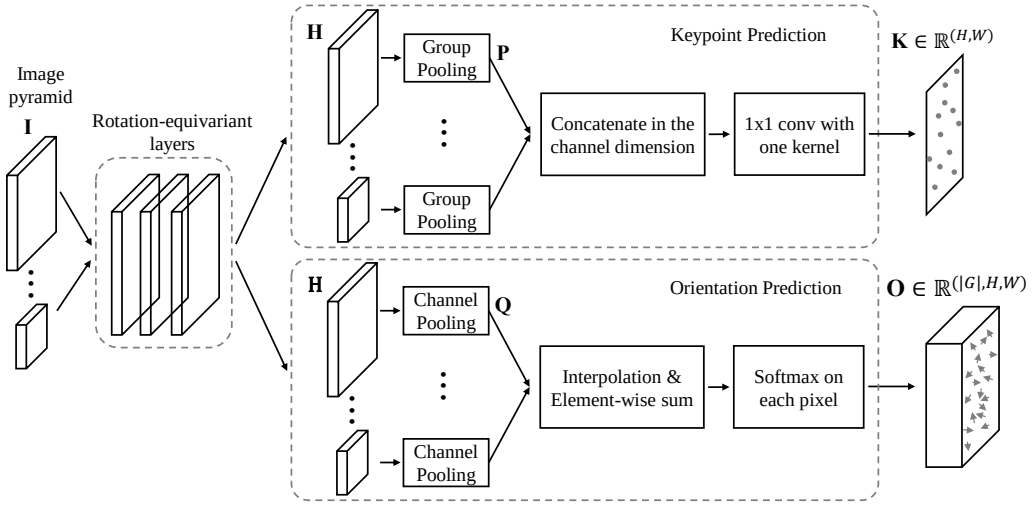

Figure 1: Overall architecture. The rotation-equivariant convolutional block takes an input image and processes it at multiple scales. The multi-scale rotation-equivariant representation $\mathbf{H}$ is fed into two separate branches that predict a keypoint map $\mathbf{K}$ and an orientation map $\mathbf{O}$.

3.2 PRELIMINARIES

Equivariance. A feature extractor Φ is said to be equivariant to a geometric transformation g if transforming an input x by the transformation g and then passing it through the feature extractor Φ gives the same result as first mapping x through Φ and then transforming the feature map by g (Weiler & Cesa, 2019). Formally, the equivariance can be expressed for transformation group G and $\Phi : X \rightarrow Y$ as

$$\Phi[T_g(x)] = T'_g[\Phi(x)], \tag{1}$$

where T_g and T'_g represent transformations on each space as a predefined group action. In this case, the function Φ operates a "structure-preserving" mapping from one representation to another.

For example, convolutional operation is designed to be translation-equivariant. If T_t is a translation group $(\mathbb{R}^2, +)$, and f is the K-dimension feature mapping sent to $\mathbb{Z}^2 \rightarrow \mathbb{R}^K$, the translation equivariance can be expressed as follows:

$$[T_t f] * \psi(x) = [T_t[f * \psi]](x), \tag{2}$$

where ψ denotes convolution filter weights $\mathbb{Z}^2 \rightarrow \mathbb{R}^K$, $*$ indicates convolution operation.

Group-equivariant convolution. Recent studies (Weiler & Cesa, 2019; Cohen et al., 2019; Weiler et al., 2018; Cohen & Welling, 2016a;b) have developed convolutional neural networks that are equivariant to symmetry groups of translation, rotation and reflection. Let H be a rotation group. For example, the cyclic group C_N represents an interval of $2\pi/N$ representing discrete rotations. The group G can be defined by $G \cong (\mathbb{R}^2, +) \rtimes H$ as the semidirect product of the translation group $(\mathbb{R}^2, +)$ with the rotation group H. Then, the rotation equivariant convolution on group G can be defined as:

$$[T_g f] * \psi(g) = [T_g[f * \psi]](g), \tag{3}$$

by replacing $x \in (\mathbb{R}^2, +)$ with $g \in G$ in Eq. 2. This operation can apply to an input tensor to produce a translation and rotation equivariant output. A rotation-equivariant network can be constructed by stacking rotation-equivariant layers similar to standard CNNs. This network becomes equivariant to both translation and rotation in the same way with the translation-equivariant convolutional networks. Formally, let $\Phi = \{L_i | i \in \{1, 2, 3, ...M\}\}$, which consists of M rotation-equivariant layers under group G. For one layer $L_i \in \Phi$, the transformation T_g is defined as

$$L_i[T_g(g)] = T_g[L_i(g)], \tag{4}$$

which indicates that the output is preserved after L_i about T_g. Extending this, if we apply T_g to input I and then pass it through the network ϕ, the transformation T_g is preserved for the whole network.

$$[\Pi_{i=1}^{M} L_i](T_g I) = T_g[\Pi_{i=1}^{M} L_i](I). \tag{5}$$

3.3 Rotation-equivariant keypoint networks

In this subsection, we describe the proposed rotation-equivariant keypoint network.

Rotation-equivariant feature extraction. For feature extraction, we use the E(2)-equivariant convolutional layers (Weiler & Cesa, 2019). For computational efficiency in a limited computational resource, we consider a discrete rotation group only. The layer acts on $(\mathbb{R}^2, +) \rtimes C_N$ and is equivariant for all translations and N discrete rotations. Given an input image, M stacked layers produce an output feature map via

$$\mathbf{H} = [\Pi_{i=1}^{M} L_i](T_g I), \tag{6}$$

where $\mathbf{H} \in \mathbb{R}^{|G| \times C \times H \times W}$ is a rotation-equivariant representation output, and C is the number of channels assigned for each group. In our experiments, we use 3 layers ($M = 3$). The output $\mathbf{H} \in \mathbb{R}^{|G| \times C \times H \times W}$ is a group of feature maps, which represents C-channel feature maps for $|G|$ orientations, and $\mathbf{H}_i$ denotes a feature map for ith orientation in G. This rotation-equivariant network enables an extensive sharing of kernel weights for different orientations, i.e., rotation transformations, and thus increasing sample efficiency in learning, particularly a rotation-involving task.

Rotation-invariant keypoint detection. Robust keypoints need to be invariant to rotation transformations; the keypointness, i.e., keypoint score, for a specific position on an image should not be affected by rotating the image. To obtain such a rotation-invariant map for keypoint scores, we collapse the group G of $\mathbf{H} \in \mathbb{R}^{|G| \times C \times H \times W}$ by group pooling, reducing it to a rotation-invariant representation $\mathbf{P} \in \mathbb{R}^{C \times H \times W}$. Specifically, we use max pooling over orientations: $\mathbf{P} = \max_g \mathbf{H}_{g,:,:,:}$. Given multi-scale outputs $\{\mathbf{P}_s\}_{s \in S}$, the final score map $\mathbf{K} \in \mathbb{R}^{H \times W}$ is obtained using standard convolution ρ over a concatenation of $\mathbf{P}_s$:

$$\mathbf{K} = \rho(\bigcup_{s \in S} (\zeta(\mathbf{P}_s))), \tag{7}$$

where ρ is a convolution operation, $\bigcup$ means concatenation of the elements, and ζ denotes a bilinear interpolation function. The interpolation function resizes the input map to target size, and the convolution transforms a rotation-invariant feature map to a rotation-invariant score map.

Rotation-equivariant orientation estimation. To estimate a characteristic orientation for a candidate keypoint, we leverage the orientation group of rotation-equivariant tensor $\mathbf{H}$ and translate it to the orientation histogram tensor $\mathbf{Q}$. Specifically, we collapse the channel dimension C for each orientation by channel pooling and produce a $|G|$-channel feature map $\mathbf{Q} \in \mathbb{R}^{|G| \times H \times W}$, where each position can be seen as being assigned an orientation histogram of $|G|$ bins. We use the implementation with 1×1 group convolution with a single filter to collapse the channels of each orientation:

$$\mathbf{Q} = \eta(\mathbf{H}_{:,c}), \tag{8}$$

where $\eta : \mathbb{R}^{|G| \times C} \rightarrow \mathbb{R}^{|G|}$ maps $\mathbf{H}$ to a discrete histogram distribution of $|G|$ bins, which is implemented by 1×1 group convolution. Note that the channel pooling can be any other operations, e.g., max pooling, average pooling, and so on. The resultant output can be interpreted as a map of characteristic orientations for corresponding positions. The output pixel-level rotation-equivariant representation $\mathbf{Q}$ is used to learn the keypoint orientation as a histogram-based dense probability map. Given multi-scale outputs $\{\mathbf{Q}_s\}_{s \in S}$, the final orientation probability tensor $\mathbf{O} \in \mathbb{R}^{|G| \times H \times W}$ is obtained by summing the outputs over the multiple scales.

$$\mathbf{O} = \sigma(\bigoplus_{s \in S}(\zeta(\mathbf{Q}_s))), \tag{9}$$

where $\sigma \in \mathbb{R}^{|G|} \rightarrow [0, 1]^{|G|}$ is a softmax function, and $\bigoplus$ is element-wise summation operation.

3.4 Training

In this subsection, we describe a loss function for the keypoint detection and the loss for the characteristic orientation for the keypoints. First, the orientation learning method will be described.

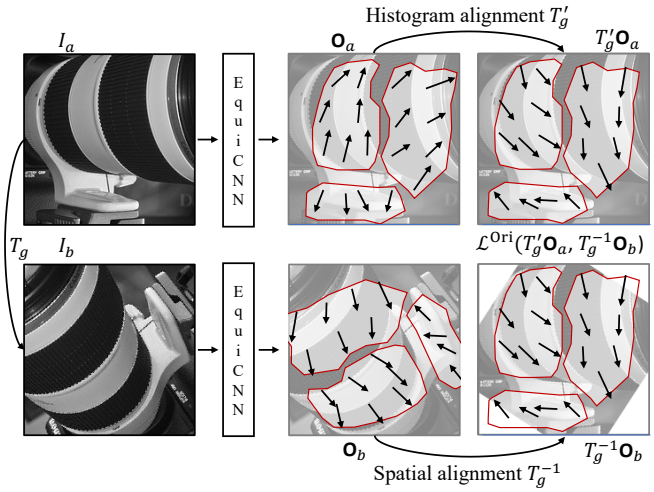

Figure 2: Illustration of orientation alignment loss. The dense orientation histogram $\mathbf{O}_\text{b}$ is spatially aligned using T_g^{-1}. The equivariant histogram vectors of the feature points in $\mathbf{O}_\text{a}$ are shifted using T'_g. The out-of-plane regions are excluded when computing the loss.

Orientation alignment loss. We train the histogram-based representation to assign orientations to keypoints. Our method takes the advantages of both assigning orientation to keypoints with the existing image histogram-based method, e.g., aggregating image gradients (Lowe, 2004), and the learning-based methods (Yi et al., 2016b; Ono et al., 2018). The dense orientation tensor $\mathbf{O} \in \mathbb{R}^{|G| \times H \times W}$ described in Sec. 3.3 encodes relative orientations for each feature point. We use an alignment technique to explicitly learn a discriminative orientation representation. Image pair I_a and I_b are used to input with the known ground-truth rotation T_g. First, we rotate the feature representation $\mathbf{O}_\text{b}$ for the given ground-truth rotation T_g^{-1} for spatially alignment. Next, histogram alignment is performed by shifting the equivariant histogram (in cyclic group G) on $\mathbf{O}_\text{a}$ using the relative shifting T'_g. The out-of-bound regions due to rotation are masked with 0. Finally, the two aligned feature pair is trained with the following cross-entropy form loss:

$$\mathcal{L}^{\text{ori}} = -\sum_{i=1}^{W}\sum_{j=1}^{H}\mathbf{M} \cdot \sum_{k=1}^{|G|} T'_g(\mathbf{O}_\text{a}) \log(T_g^{-1}(\mathbf{O}_\text{b})), \tag{10}$$

where $\mathbf{M} = \mathbf{1} \wedge T_g^{-1}(\mathbf{1})$ with $\mathbf{1} \in 1^{H \times W}$ filled in 1 is mask map which is out-of-bound due to the rotation. We omit the index i, j of the tensor $\mathbf{M}$, and i, j, k of $\mathbf{O}_\text{a}$ and $\mathbf{O}_\text{b}$ in Eq. 10 for

simplicity. A pixel-level aligned value $\mathcal{L}^{\mathrm{ori}}_{i,j}$ is a similarity score for how well the orientation of the pixel corresponding to I_{a} and I_{b} is predicted. Note that $\mathcal{L}^{\mathrm{ori}}_{i,j}$ is spatially aligned of coordinate (i, j) in I_{a}. Figure 2 shows the illustration of our loss function.

Window-based keypoint detection loss. We utilize the window-based keypoint estimation (Barroso-Laguna et al., 2019). In the case of keypoints, the ground truth is not well defined. However, in general, a good keypoint means to be extracted a consistent location invariant to geometric or photometric image transformations. Some researches (Lenc & Vedaldi, 2016; Ono et al., 2018) design a loss function to obtain keypoints consistent for homography transformation, and other researches (DeTone et al., 2018; Verdie et al., 2015; Zhang et al., 2017) obtain keypoints based on anchors. Barroso-Laguna et al. (2019) propose to select anchor-based keypoints, and then take the corresponding region using the ground-truth homography as the inputs of the loss function. We use the loss function of Barroso-Laguna et al. (2019) which takes advantage of both anchor-based methods and homography consistency.

The keypoint detection loss uses multi-scale windows based on the index proposal. The keypoint score map $\mathbf{K} \in \mathbb{R}^{H \times W}$ is transformed by non-maximal suppression through exponential scaling based on window as follows:

$$m^{(i)}_{u,v} = \frac{e^{w^{(i)}_{u,v}}}{\sum_{j=1}^{N}\sum_{k=1}^{N} e^{w^{(i)}_{j,k}}}, \tag{11}$$

where a window $w^{(i)}$ is $N \times N$ grid in score map $\mathrm{ReLU}(\mathbf{K})$ with the score value at each index (u, v). $m^{(i)}$ is non-maximal suppressed $N \times N$ patch. Then the max value in $m^{(i)}$ becomes dominant on the window, and a weighted average using the index kernel is performed to obtain real value coordinates.

$$[x^{(i)}, y^{(i)}]^T = [\bar{u}^{(i)}, \bar{v}^{(i)}]^T = \sum_{u=1}^{N}\sum_{v=1}^{N} [W_u \odot m^{(i)}_{u,v}, W_v \odot m^{(i)}_{u,v}]^T + c_{w^{(i)}}. \tag{12}$$

where $\odot$ is a pointwise product, W is a $N \times N$ index kernel, and c_w is the top-left coordinates of window $w^{(i)}$. This soft index selection method makes it possible to have real-valued coordinates and differentiable, unlike the $\arg\max$ function. To give the estimated keypoints coordinates covariant properties to geometric transformations, index proposal loss is used as:

$$\mathcal{L}^{\mathrm{kpts}}(I_{\mathrm{a}}, I_{\mathrm{b}}, T_g, N) = \sum_i \alpha^{(i)} || [x^{(i)}, y^{(i)}]^T_a - T_g^{-1} [\hat{x}^{(i)}, \hat{y}^{(i)}]^T_b ||^2, \tag{13}$$

where $[\hat{x}^{(i)}, \hat{y}^{(i)}]$ is weighted average index without exponential on window and $[x^{(i)}, y^{(i)}]$ are soft-selected scores using Eq. 12, and T_g^{-1} is the ground-truth geometric transformation I_{b} to I_{a}. $\alpha^{(i)} = K_{y^{(i)},x^{(i)};a} + T_g^{-1} K_{\hat{y}^{(i)},\hat{x}^{(i)};b}$ is weighting term using keypoint score maps. Finally, we use multiple window sizes with switching term of the source and target images as in Barroso-Laguna et al. (2019):

$$\mathcal{L}^{\mathrm{kpts}}(I_{\mathrm{a}}, I_{\mathrm{b}}, H_{a,b}) = \sum_l \lambda_l (\mathcal{L}^{\mathrm{kpts}}(I_{\mathrm{a}}, I_{\mathrm{b}}, T_g, N_l) + \mathcal{L}^{\mathrm{kpts}}(I_{\mathrm{b}}, I_{\mathrm{a}}, T_g^{-1}, N_l)), \tag{14}$$

where l is the index of scale level, N_l is the window size, λ_l is the balancing parameter at scale level. We use the final loss function as follows:

$$\mathcal{L} = \alpha \mathcal{L}^{\mathrm{ori}} + \mathcal{L}^{\mathrm{kpts}}, \tag{15}$$

where α is balancing parameter of the loss functions.

4 EXPERIMENTS

This section describes comparative experiments to demonstrate the effectiveness of our model. In section 4.1, we describe the implementation details and experimental benchmarks. In section 4.2, we show the results of keypoint detection and matching compared to existing keypoint detection methods. In section 4.3, we show the results of 6 DoF pose estimation for the transferability. In section 4.4, we additionally verify the effectiveness of our model by changing group size, replacing rotation-equivariant CNNs to regular CNNs, and visualizing the output representations and matches.

4.1 EXPERIMENTAL SETTING

Implementation details. We use the $E(2)$-CNN framework (Weiler & Cesa, 2019) for the implementation with PyTorch. For training, we use siamese networks, and input pairs share the weights updated at the same time. We use the cyclic group G size 36, with the channel dimension C size 2. We use a equivariant backbone with 3-layers, each of layer consists of a `conv-bn-relu` module. Each convolution layer has 5×5 kernel with padding 2 without bias, and model parameters are randomly initialized. We use a batch size of 16. We train with Adam optimizer with a learning rate of 0.001. The leaning rate decay is 0.5 every 10 epochs in total 20 epochs. We use the same configuration of the loss $\mathcal{L}^{\text{kpts}}$ of Barroso-Laguna et al. (2019), and the loss balancing parameter α is 100. In all the experiments, we use a Intel Xeon Gold 6240 CPU running at 2.60GHz and an NVIDIA GeForce RTX 3090.

Training dataset. We use a synthetic dataset for self-supervised training. Our model needs a ground-truth relative orientation for training. Although the existing dataset, such as the HPatches, has a homography matrix, the exact orientation is not uniquely defined. Therefore, we define relative rotation uniquely by composing synthetic affine transformation with separate affine parameters. We tried to train our model using the random scale, skew, and rotation parameters. However, the model trained only with rotation shows better transferability to datasets with various geometric transformations, so we generate randomly to transform with rotation [-180, 180]. To improve the robustness of illumination changes, we modify the contrast, brightness, and hue value in HSV space. We exclude the images with insufficient edges through Sobel filters as a pre-processing. The synthetic dataset has 9,100 image pairs of size 192×192 split into 9,000 as a training set and 100 as a validation set by using the ILSVRC2012 (Russakovsky et al., 2015) as the source data.

Evaluation benchmark. We use two evaluation benchmarks: HPatches and IMC2021 (Balntas et al., 2017; Jin et al., 2021). The HPatches is for evaluating keypoint detection and matching. IMC2021 is used for transferability to complex tasks by measuring the SfM quality using the 6 DoF pose estimation accuracy.

HPatches consists of 116 scenes with 59 viewpoint variation and 57 illumination variation (Balntas et al., 2017). Each scene consists of 5 image pairs with ground-truth planar homography, a total of 696 image pairs. We compare our model with the existing models using 1,000 keypoints for evaluation. We use the repeatability score, the number of matches, and mean matching accuracy (MMA) as evaluation metrics proposed to (Mikolajczyk & Schmid, 2005; Lenc & Vedaldi, 2018; Zhang et al., 2017). Repeatability score is the ratio between the number of repeatable keypoints over the total number of detections. MMA is the average percentage of correct matches per image pair. We use the thresholds 3 pixel and 5 pixel to measure correct matches.

IMC2021 is a large-scale challenge dataset of wide-baseline matching (Jin et al., 2021). IMC2021 consists of an unconstrained urban scene with large illumination and viewpoint variations. In this experiment, we compare the effect of keypoint detection methods in the image matching pipeline Mishchuk et al. (2017); Cavalli et al. (2020); Chum et al. (2005). We experiment on the stereo track using a Phototourism and PragueParks validation set. This benchmark takes matches as input and measures 6 DoF pose estimation accuracy. We measure the mean average accuracy (mAA) of pose estimation at 5°and 10°and the number of inliers.

Outlier rejection with keypoint orientation. We conduct outlier rejection to show the effectiveness of our orientation with predicted matches. To fully utilize our equivariant orientation representation, we introduce a simple outlier rejection method using our equivariant orientation representation $\mathbf{O}$. We compute the difference of estimated orientation for tentative matches and then derive the most frequent difference between two images. We exclude matches far from the most frequent difference as the outlier. The inliers $\mathbf{p}$ is defined as follows:

$$\mathbf{p}_i(\mathbf{O}_a, \mathbf{O}_b, t) = \begin{cases} \text{inlier}, & \text{if } |\texttt{mode}(\mathbf{d}) - \mathbf{d}_i| \leq t, \\ \text{outlier}, & \text{otherwise}, \end{cases} \tag{16}$$

where the difference of orientation vector $\mathbf{d} = (\mathbf{o}_b - \mathbf{o}_a + 360) \bmod 360$, the assigned orientation $\mathbf{o} = \arg\max_g \delta(\mathbf{O})_g$, $\delta : \mathrm{R}^{|G| \times H \times W} \rightarrow \mathrm{N}^{|G| \times K}$ selects the keypoint coordinates by matcher, K is the number of matches, t is a threshold, `mode` operation returns the most frequent value on the input vector. We use the outlier threshold $t = 30$.

Det.	Desc.	All variations			
		Rep.	MMA		pred.
			@3px	@5px	match.
SIFT	SIFT	41.9	49.4	52.4	404.2
SIFT	HardNet	41.9	57.1	62.3	437.8
Key.Net	HardNet	55.9	72.5	79.4	474.4
ours	HardNet	**56.0**	69.9	78.6	**522.1**
ours*	HardNet	**56.0**	**74.8**	**82.3**	443.9
SIFT	SOSNet	41.9	57.9	63.0	430.8
Key.Net	SOSNet	55.9	72.7	79.6	464.7
ours	SOSNet	**56.0**	70.4	79.2	**514.8**
ours*	SOSNet	**56.0**	**75.2**	**82.7**	440.1
SIFT	HyNet	41.9	57.3	62.5	438.9
Key.Net	HyNet	55.9	72.0	78.9	475.3
ours	HyNet	**56.0**	69.8	78.6	**522.1**
ours*	HyNet	**56.0**	**75.0**	**82.5**	442.5

Table 1: Results on the HPatches. 'Det.' denotes keypoint detection method, 'Desc.' denotes descriptor extraction method, 'Rep.' denotes the repeatability score, and 'pred. match.' is the average number of predicted matches. '*' besides ours means the outlier filtering method using our orientation. Numbers in bold indicate the best scores.

Det.	K	Stereo track.		
		Num. Inl.	mAA(5°)	mAA(10°)
DoG+AN	1,024	43.8	0.210	0.277
Key.Net	1,024	100.6	0.345	0.447
ours	1,024	**131.2**	**0.403**	**0.522**
DoG+AN	2,048	105.9	0.385	0.477
Key.Net	2,048	217.8	0.452	0.568
ours	2,048	**250.5**	**0.460**	**0.581**

Table 2: Mean average accuracy (mAA; 5°, 10°) of 6-DoF pose estimation and the number of inlier matches (Num. Inl.) on IMC2021 validation set (Jin et al., 2021). Column 'K' denotes the number of keypoints.

Results on the HPatches. Table 1 shows the results of keypoint detection and keypoint matching in HPatches (Balntas et al., 2017). We compare a handcrafted model SIFT, and a learning-based model Key.Net as baseline keypoint detectors (Lowe, 2004; Barroso-Laguna et al., 2019), and with patch-based descriptor extraction methods, HardNet, SOSNet, and HyNet (Mishchuk et al., 2017; Tian et al., 2019; 2020). We use the mutual nearest neighbor matching algorithm for all cases in this experiment. Our model improves the repeatability score, which is an evaluation metric for keypoint detection. This shows that our equivariant representation detects more consistent keypoints than SIFT and Key.Net for viewpoint & illumination changes. Even though our results without outlier rejection have lower MMA than Key.Net, the number of predicted matches is larger, which means the actual number of correct matches is higher than Key.Net. Furthermore, our model using the outlier rejection consistently obtains better MMAs than the baseline models. This shows that our orientation is effective in keypoint matching on this large viewpoint & illumination variation dataset.

Results on the IMC2021. Table 2 shows the results of 6 DoF pose estimation in IMC2021 for measuring transferability. For this experiment, we use the rest of the image matching pipeline using HardNet descriptor (Mishchuk et al., 2017), and DEGENSAC geometric verification Chum et al. (2005) with AdaLAM (Cavalli et al., 2020) for all cases. We compare to two baselines, DoG+AN (Lowe, 2004; Mishkin et al., 2018) and Key.Net (Barroso-Laguna et al., 2019). We use our keypoint detector without the orientation to compare the effect of the keypoint detector. The result shows that our model consistently improve the camera pose estimation accuracy (mAAs) and the number of inliers compared to the baseline models on this complex tasks of general scenes. We evaluate to use the provided source code from IMC2021[1].

4.2 Additional results

Effect of orientation estimation. Table 3 shows the comparison with an orientation estimation method (Lowe, 2004) based on the image gradient. Key.Net (Barroso-Laguna et al., 2019) are excluded because they do not generate orientation. The result of our orientation (Row 3) yields higher MMA than the result of SIFT orientation (Row 2) with SIFT keypoints. This shows that our orientation is more effective than the orientation based on image gradients.

Results of different group size and without equivariant layers. Table 4 shows the results of the matching scores with the number of parameters according to group size. We make the same computation of all models by changing the channel C. Therefore, the model size increases by N times whenever the group size decreases by N times. For example, Row 3 with the group size 9 has

[1]https://github.com/ubc-vision/image-matching-benchmark

Det.	Ori.	Rej?	MMA @3px	MMA @5px	match.
SIFT	SIFT		49.4	52.4	404.2
SIFT	SIFT	✓	52.6	55.8	**251.6**
SIFT	ours	✓	**63.7**	**67.4**	236.5

Table 3: Comparison of the orientation estimation on the HPatches. 'Det.' denotes the keypoint detector, 'Ori.' denotes the orientation estimation method, and 'Rej?' denotes whether or not to use outlier rejection. We use the SIFT descriptor in all cases.

	MMA w/o out. rej. @3px	MMA w/o out. rej. @5px	MMA out. rej. @3px	MMA out. rej. @5px	# param.
G_{36}	**69.9**	**78.6**	**74.8**	**82.3**	**3.3K**
G_{18}	66.2	75.0	72.7	80.8	6.5K
G_9	62.4	70.7	72.0	79.1	13.0K
G_8	63.2	73.7	69.5	79.0	14.7K
G_4	62.3	70.7	68.2	75.8	29.1K
-	64.5	74.0	64.5	74.0	116K

Table 4: Experiment according to group size change. The subscript of G denotes the group size. 'out. rej.' denotes the results with outlier rejection.

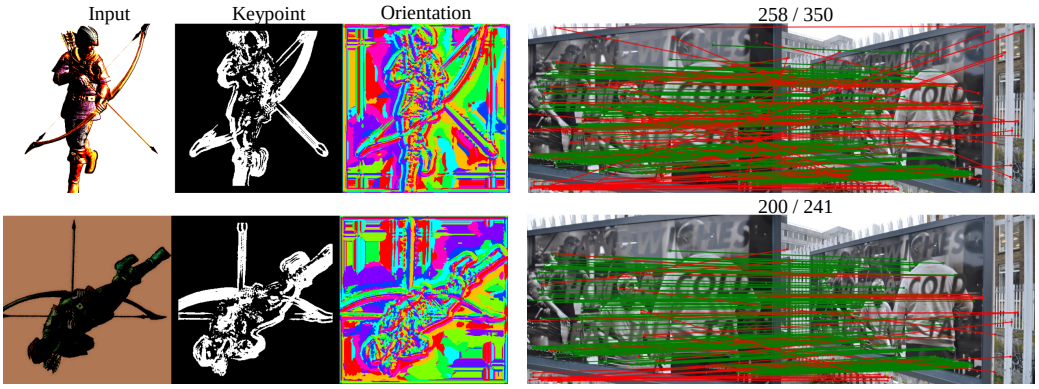

Figure 3: Qualitative results. The left sides are visualizations of keypoint score map and color-coded orientation map by $\arg\max_g \mathbf{O}_g$. The top is the source image, and the bottom is the target image. The top right is the visualization of keypoint matching using our keypoints with HardNet, and the bottom right is the matches with outlier rejection using our orientation. We map the orientation range from $[0, 359)$ to $[0, 255)$ to visualize the estimated orientation by hue of HSV color representation. We use 3 pixel threshold of the correct match.

the number of channel 8. In the table, the result of group size 36 is the best with the smallest model size. The last row, which replaces rotation-equivariant layers with regular convoluational layers, has a large number of parameters because there is no weight sharing. In addition, the model with regular convoluational layers fails to train orientation, so the outlier rejection has no effect. These results show that as the number of groups increases, the number of parameters can be significantly reduced without losing performance. Furthermore, group-equivariant convolutional operations are important for orientation learning.

Qualitative results. Figure 3 visualizes score maps and keypoint matching. The keypoint score map on the second column of the left side shows that our model consistently finds keypoints invarint to rotation. The orientation map on the third column of the left side shows that the modality of the orientation of an pixel consistently changes as it is rotated. The result on the right side shows outlier rejection using our orientatino effectively removes false positives.

5 Conclusion

This paper presents a self-supervised keypoint detection method using rotation-equivariant CNNs. The rotation-equivariant representation generates rotation-invariant keypoints and rotation-equivariant orientations. We propose the orientation alignment loss to predict dense orientation using the rotation-equivariant representation. The output orientation combined with the outlier rejection performs better matching accuracy than the existing keypoint detection methods. Our keypoint detector transfers well on the more complex wide-baseline image matching task. In the future, this study can be extended to various geometric transformation groups, e.g., affine/non-rigid, to overcome our model limits to the cyclic rotation group. We leave this for the future.

CHECKLIST

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section **??**.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [No]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] , We plan to release source code after paper acceptance.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

We take this format from the NeurIPS 2021 author guideline.

REFERENCES

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*, pp. 5173–5182, 2017.

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*, pp. 5836–5844, 2019.

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

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

Christopher B Choy, JunYoung Gwak, Silvio Savarese, and Manmohan Chandraker. Universal correspondence network. pp. 2414–2422, 2016.

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, pp. 772–779. IEEE, 2005.

Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pp. 2990–2999. PMLR, 2016a.

Taco S Cohen and Max Welling. Steerable cnns. *arXiv preprint arXiv:1612.08498*, 2016b.

Taco S Cohen, Mario Geiger, and Maurice Weiler. A general theory of equivariant cnns on homogeneous spaces. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pp. 9145–9156, 2019.

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*, pp. 224–236, 2018.

Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-net: A trainable cnn for joint description and detection of local features. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pp. 8092–8101, 2019.

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*, pp. 253–262, 2019.

Jiaming Han, Jian Ding, Nan Xue, and Gui-Song Xia. Redet: A rotation-equivariant detector for aerial object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2786–2795, 2021.

Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *Advances in neural information processing systems*, 28:2017–2025, 2015.

Wei Jiang, Eduard Trulls, Jan Hosang, Andrea Tagliasacchi, and Kwang Moo Yi. Cotr: Correspondence transformer for matching across images. *arXiv preprint arXiv:2103.14167*, 2021.

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.

Seungryong Kim, Stephen Lin, Sangryul Jeon, Dongbo Min, and Kwanghoon Sohn. Recurrent transformer networks for semantic correspondence. *arXiv preprint arXiv:1810.12155*, 2018.

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*, pp. 898–908, 2021.

Karel Lenc and Andrea Vedaldi. Learning covariant feature detectors. In *European conference on computer vision*, pp. 100–117. Springer, 2016.

Karel Lenc and Andrea Vedaldi. Large scale evaluation of local image feature detectors on homography datasets. *arXiv preprint arXiv:1807.07939*, 2018.

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

Diego Marcos, Michele Volpi, Nikos Komodakis, and Devis Tuia. Rotation equivariant vector field networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5048–5057, 2017.

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

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

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

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

Juhong Min, Jongmin Lee, Jean Ponce, and Minsu Cho. Hyperpixel flow: Semantic correspondence with multi-layer neural features. 2019.

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*, pp. 4826–4837, 2017.

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)*, pp. 284–300, 2018.

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*, pp. 3456–3465, 2017.

David Novotny, Diane Larlus, and Andrea Vedaldi. Anchornet: A weakly supervised network to learn geometry-sensitive features for semantic matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5277–5286, 2017.

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

Nicolas Pielawski, Elisabeth Wetzer, Johan Öfverstedt, Jiahao Lu, Carolina Wählby, Joakim Lindblad, and Nataša Sladoje. CoMIR: Contrastive multimodal image representation for registration. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 18433–18444. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/d6428eecbe0f7dff83fc607c5044b2b9-Paper.pdf.

Jerome Revaud, Cesar De Souza, Martin Humenberger, and Philippe Weinzaepfel. R2d2: Reliable and repeatable detector and descriptor. *Advances in neural information processing systems*, 32: 12405–12415, 2019.

Ignacio Rocco, Relja Arandjelovic, and Josef Sivic. Convolutional neural network architecture for geometric matching. 2017.

Ignacio Rocco, Relja Arandjelović, and Josef Sivic. End-to-end weakly-supervised semantic alignment. 2018a.

Ignacio Rocco, Mircea Cimpoi, Relja Arandjelović, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Neighbourhood consensus networks. pp. 1656–1667, 2018b.

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

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

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Paul Hongsuck Seo, Jongmin Lee, Deunsol Jung, Bohyung Han, and Minsu Cho. Attentive semantic alignment with offset-aware correlation kernels. 2018.

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*, pp. 8132–8140, 2019.

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

Ivan Sosnovik, Michał Szmaja, and Arnold Smeulders. Scale-equivariant steerable networks. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=HJgpugrKPS`.

Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8922–8931, 2021.

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*, pp. 661–669, 2017.

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*, pp. 11016–11025, 2019.

Yurun Tian, Axel Barroso Laguna, Tony Ng, Vassileios Balntas, and Krystian Mikolajczyk. Hynet: Learning local descriptor with hybrid similarity measure and triplet loss. *Advances in Neural Information Processing Systems*, 33, 2020.

Prune Truong, Martin Danelljan, Luc V Gool, and Radu Timofte. Gocor: Bringing globally optimized correspondence volumes into your neural network. *Advances in Neural Information Processing Systems*, 33, 2020a.

Prune Truong, Martin Danelljan, and Radu Timofte. Glu-net: Global-local universal network for dense flow and correspondences. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6258–6268, 2020b.

Yannick Verdie, Kwang Yi, Pascal Fua, and Vincent Lepetit. Tilde: A temporally invariant learned detector. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5279–5288, 2015.

Maurice Weiler and Gabriele Cesa. General e (2)-equivariant steerable cnns. *Advances in Neural Information Processing Systems*, 32:14334–14345, 2019.

Maurice Weiler, Fred A Hamprecht, and Martin Storath. Learning steerable filters for rotation equivariant cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 849–858, 2018.

Daniel E Worrall, Stephan J Garbin, Daniyar Turmukhambetov, and Gabriel J Brostow. Harmonic networks: Deep translation and rotation equivariance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5028–5037, 2017.

Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. Lift: Learned invariant feature transform. In *European conference on computer vision*, pp. 467–483. Springer, 2016a.

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*, pp. 107–116, 2016b.

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*, pp. 6818–6826, 2017.

Yanzhao Zhou, Qixiang Ye, Qiang Qiu, and Jianbin Jiao. Oriented response networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 519–528, 2017.