Steerers: A framework for rotation equivariant keypoint descriptors

Georg Bökman[†] Johan Edstedt[‡] Michael Felsberg[‡] Fredrik Kahl[†]
[†]Chalmers University of Technology [‡]Linköping University

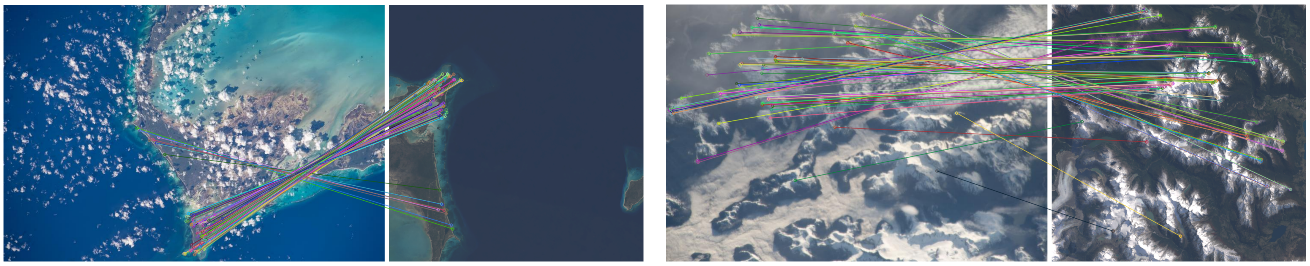

Figure 1. **Matching under large in-plane rotations.** Two challenging pairs from AIMS [49]. The left images in each pair were taken by astronauts on the ISS and are geo-referenced by matching them with the satellite images on the right. We plot estimated inlier correspondences after homography estimation with RANSAC. Further qualitative examples are shown in the appendix.

Abstract

Image keypoint descriptions that are discriminative and matchable over large changes in viewpoint are vital for 3D reconstruction. However, descriptions output by learned descriptors are typically not robust to camera rotation. While they can be made more robust by, e.g., data augmentation, this degrades performance on upright images. Another approach is test-time augmentation, which incurs a significant increase in runtime. Instead, we learn a linear transform in description space that encodes rotations of the input image. We call this linear transform a steerer since it allows us to transform the descriptions as if the image was rotated. From representation theory, we know all possible steerers for the rotation group. Steerers can be optimized (A) given a fixed descriptor, (B) jointly with a descriptor or (C) we can optimize a descriptor given a fixed steerer. We perform experiments in these three settings and obtain state-of-the-art results on the rotation invariant image matching benchmarks AIMS and Roto-360. We publish code and model weights at this https url.

1. Introduction

Discriminative local descriptions are vital for multiple 3D vision tasks, and learned descriptors have recently been shown to outperform traditional handcrafted local features [17, 19, 23, 43]. One major weakness of learned descriptors compared to handcrafted features such as SIFT [35] is the relative lack of robustness to non-upright images [55]. While images taken from ground level can sometimes be made upright by aligning with gravity as the canonical orientation, this is not always possible. For example, descriptors robust to rotation are vital in space applications [49], as well as medical applications [42], where no such canonical orientation exists. Even when a canonical orientation exists, it may be difficult or impossible to estimate. Rotation invariant matching is thus a key challenge.

The most straightforward manner to get rotation invariant matching is to train or design a descriptor to be rotation invariant [17, 35]. However, this sacrifices distinctiveness in matching images with small relative rotations [41]. An alternative approach is to train a rotation-sensitive descriptor and perform test-time-augmentation, selecting the pair that produces the most matches. The obvious downside of TTA is computational cost. For example, testing all 45° rotations requires running the model eight times.

In this paper, we present an approach that maintains distinctiveness for small rotations and allows for rotation invariant matching when we have images with large rotations. We do this while adding only negligible additional runtime, running the descriptor only a single time. The main idea is to learn a linear transform in description space that corresponds to a rotation of the input image; see Figure 2. We call this linear transform a *steerer* as it allows us to modify keypoint descriptions as if they were describing rotated images—we can *steer* the descriptions without having to rerun the descriptor network. We show empiri-

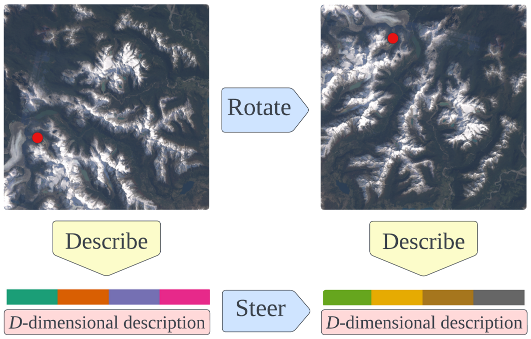

Figure 2. **Overview of approach.** A steerer (Definition 4.4) is a linear map that transforms the description of a keypoint into the description of the corresponding keypoint in a rotated image. Thus, a steerer makes the keypoint descriptor rotation equivariant, and we can obtain the descriptions of keypoints in arbitrarily rotated images while only running the descriptor once.

cally that approximate steerers can be obtained for existing descriptors and motivate this theoretically. We also investigate jointly optimizing steerers and descriptors and show how this enables nearly exact steering while not sacrificing performance on upright images. Using mathematical representation theory, we can describe all possible steerers—they are representations of the rotation group. This enables the choice of a fixed steerer and training a descriptor for it, and in turn, the investigation of which steerers give the best performance.

Using our framework, we set a new state-of-the-art on the rotation invariant matching benchmarks AIMS [49] (Figure 1) and Roto-360 [31]. At the same time, we are with the same models able to perform on par with or even outperform existing non-invariant methods on upright images on the competitive MegaDepth-1500 benchmark [33, 50].

In summary, our main contributions are as follows.

1. We introduce a new framework of steerers for equivariant keypoint descriptors (Section 4) and theoretically motivate why steerers emerge in practice (Section 5).
2. We develop several settings for investigating steerers (Section 5.1) and ways to apply them for rotation invariant matching (Section 5.2).
3. We conduct a large set of experiments, culminating in state-of-the-art on AIMS and Roto-360 (Section 6).

2. Related work

Classical keypoint descriptions are typically made rotation invariant by using keypoints with associated local rotation frames and computing the descriptions in these frames. Examples include SIFT [35], SURF [8], and ORB [45]. A canonical rotation frame can be used for patch-based neural network descriptors as well [53, 54]. Further, neural network-based approaches have been proposed for estimating the keypoint rotation frame [29, 30, 37] and for both computing the rotation frame and the descriptions in that frame [31, 61]. Notably, [30, 31] use rotation equivariant ConvNets [15, 56, 59]. Equivariant ConvNets have also been used for rotation-robust keypoint detection without estimating the rotation frame [2, 46], keypoint description [2, 34] and end-to-end image matching [9]. In theory[1], equivariant ConvNets guarantee that the predictions are consistent when rotating the image. They are one example of hard-coding equivariance into network layers using mathematical group theory, an idea that goes back to the 1990's [58] and has seen large recent interest [11, 20, 22].

Neural networks can also be encouraged to learn equivariance rather than having it hard-coded in the layers. This can be done by enforcing group-specific invariants in the network output space [25, 48] (see also Section 5). Another approach is to specify a group representation on the output of the network and train the network to satisfy equivariance wrt. that representation [16, 28, 36, 60]. We will use this approach for keypoint descriptions in our Setting C. The benefits of not hard-coding equivariance are that arbitrary network architectures can be used (particularly pre-trained non-equivariant networks) and that one does not need to specify the group representations acting on each layer of the network. A special case of learning equivariance is rotation invariant descriptors through data augmentation [38, 52].

A recent line of work [10, 12, 24, 32] investigates to what extent neural networks exhibit equivariance without having been trained or hard-coded to do so. They find that many networks are approximately equivariant. One major limitation is that they only consider networks trained for image classification. We will empirically demonstrate a high level of equivariance in keypoint descriptors that were not explicitly trained to be equivariant and theoretically motivate why this happens (our Setting A).

3. Preliminaries

In this work, we are interested in finding linear mappings between keypoint descriptions where the images may have been rotated independently. We will, in particular, consider the group of quarter rotations C_4 and the group of continuous rotations $\mathrm{SO}(2)$.

Ordinary typeset g will denote an arbitrary group element, boldface $\mathbf{g}$ will always mean the generator of C_4 for the remainder of the text so that the elements of C_4 are $\mathbf{g}$, $\mathbf{g}^2$, $\mathbf{g}^3$ and the identity element $\mathrm{id} = \mathbf{g}^4$. Boldface $\mathbf{i}$ will denote the imaginary unit such that $\mathbf{i}^2 = -1$. Given matrices $X_1, X_2, \dots, X_J$, the notation $\bigoplus_{j=1}^{J} X_j$ will mean the block-diagonal matrix with blocks $X_1, X_2, \dots, X_J$.

[1] It has been demonstrated that equivariant ConvNets can learn to break equivariance [18] when this benefits the task at hand. *E.g.*, the end-to-end matcher SE2-LoFTR [9] is not perfectly consistent over rotations [9, 49].

3.1. Preliminaries on keypoint matching

The underlying task is to take two images of the same scene and detect 2D points that correspond to the same 3D point. A pair of such points that depict the same 3D point is called a correspondence. The approach for finding correspondences that will be explored is a three-stage approach:

1. Detection. Detect N keypoint locations in each image.
2. Description. Describe the keypoint locations with descriptors, *i.e.*, feature vectors in $\mathbb{R}^D$.
3. Matching. Match the descriptors, typically by using mutual nearest neighbours in cosine distance.

This classical setup includes SIFT [35] but also more recent deep learning-based approaches. In particular, we follow the method in DeDoDe [19], where the keypoint detector is first optimized to find good point tracks from SfM reconstructions and the keypoint descriptor is optimized by maximizing the matching likelihood obtained by a frozen keypoint detector as follows. If the N descriptors (each normalized to unit length) in the two images are $y_1 \in \mathbb{R}^{D\times N}$ and $y_2 \in \mathbb{R}^{D\times N}$ we first form the $N \times N$ matching matrix $Y = y_1^T y_2$, and obtain a matrix of pairwise likelihoods by using the dual softmax [44, 50, 55]:

$$p(y_1, y_2) = \frac{\exp(\iota Y)}{\sum_{\text{columns}} \exp(\iota Y)} \cdot \frac{\exp(\iota Y)}{\sum_{\text{rows}} \exp(\iota Y)}. \quad (1)$$

Here $\iota = 20$ is the inverse temperature. The negative logarithm of the likelihood (1) is minimized for those pairs in the $N \times N$ matrix that correspond to ground truth inliers.

3.2. Preliminaries on group representations

Definition 3.1. (Group representation) Given a group G, a representation of G on $\mathbb{R}^D$ is a mapping $\rho : G \to \mathrm{GL}(\mathbb{R}, D)$ that preserves the group multiplication, *i.e.*, $\rho(gg') = \rho(g)\rho(g')$ for every $g, g' \in G$.

Simply stated, ρ maps every element in the group to an invertible $D \times D$ matrix. The point of using representations is that groups such as C_4 act differently on different quantities as we will illustrate in the following examples.

Example 3.1. For $\mathbb{R}^{n\times n}$ (a square image grid), C_4 can be represented by permutations of the pixels in the obvious way so that the image is rotated anticlockwise by multiples of 90°. We denote this group representation by P_{90} so that applying P_{90}^k rotates the image by $k \cdot 90°$ anticlockwise.

Example 3.2. For $\mathbb{R}^2$ (image coordinates), one possible representation of C_4 is $\rho(\mathbf{g}^k) = R_{90}^k = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}^k$. Multiplication by R_{90}^k corresponds to rotating image coordinates by $k \cdot 90°$ if the center of the image is taken as $(0, 0)$.

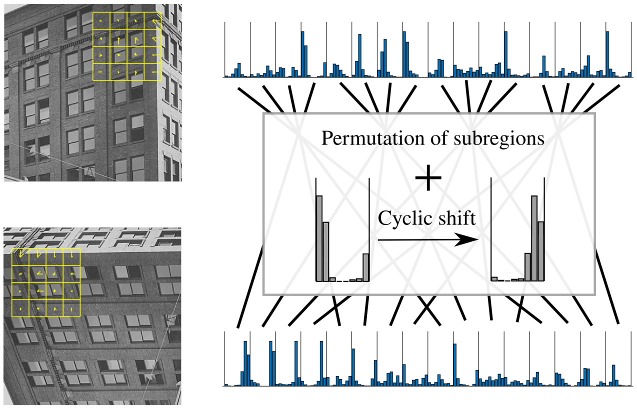

Figure 3. **Equivariance of Upright SIFT.** Left: A keypoint with its Upright SIFT description in an upright image and a rotated version. The small yellow squares are the subregions where histograms of gradient orientations are computed. Right: The Upright SIFT descriptions unravelled into the 128 bin histograms that constitute them. When we rotate the image, the subregions are permuted, and the histogram bins within each subregion are further permuted cyclically. Hence, Upright SIFT is rotation equivariant.

4. Equivariance and steerability

In this section, we will analyze the close connection between equivariance and steerability. We start with an example to introduce the former concept.

Example 4.1. SIFT descriptions [35] are 128 dim. vectors designed to be invariant to rotation, scale and illumination and highly distinctive for leveraging feature matching. For an input image $I \in \mathbb{R}^{n\times n}$ and N keypoints with scale and orientation $x \in \mathbb{R}^{4\times N}$[2], we get descriptions $y \in \mathbb{R}^{128\times N}$. If f is the SIFT descriptor, we write $f(I, x) = y$. The descriptions consist of histograms of image gradients over patches around the keypoints x. The patches are oriented by the keypoint orientations so that the descriptions are invariant to joint rotations of the image and keypoints:

$$f\left(P_{90}^k I, (\oplus_{b=1}^2 R_{90})^k x\right) = f(I, x).$$

If we discard the keypoint orientations, *i.e.*, set the angle of each keypoint to 0, we get the Upright SIFT (UPSIFT) descriptor [1, 7], which is often used for upright images as it is more discriminative than SIFT. When we rotate an image 90°, then the gradient histograms, *i.e.*, the UPSIFT descriptions are permuted by a specific permutation P_{UPSIFT}, so if f is the UPSIFT descripor, we have

$$f\left(P_{90}^k I, (\oplus_{b=1}^2 R_{90})^k x\right) = P_{\text{UPSIFT}}^k f(I, x).$$

We illustrate the permutation P_{UPSIFT} in Figure 3. UPSIFT is not rotation invariant, but it is rotation *equivariant*—when we rotate the input, the output changes predictably. Explicitly, the representation is $\rho(\mathbf{g}^k) = P_{\text{UPSIFT}}^k$.

[2] The first two coordinates of each keypoint in x are its location and the last two a vector for its orientation and scale, so x is rotated by $\oplus_{b=1}^2 R_{90}$.

Definition 4.1 (Equivariance)**.** We say that a function $f : V \to W$ is equivariant with respect to a group G if

$$\rho(g)f(v) = f(\rho_{\text{in}}(g)v), \forall v \in V, g \in G, \tag{2}$$

for some group representations ρ_{in}, ρ.

This work will mainly be concerned with the equivariance of learned keypoint descriptors of ordinary keypoints (without scale and orientation).

Definition 4.2 (Equivariance of keypoint descriptor)**.** We say that a keypoint descriptor f is equivariant with respect to a group G transforming the input image by ρ_{image} and the input keypoint locations by ρ_{keypoint} if there exists ρ such that

$$\rho(g)f(I, x) = f(\rho_{\text{image}}(g)I, \rho_{\text{keypoint}}(g)x) \tag{3}$$

for all images, keypoints and group elements. We call the descriptor invariant if $\rho(g)$ is the identity matrix for all g. Invariance is a special type of equivariance.

Example 4.2. A keypoint descriptor f is equivariant under 90° rotations if there exists ρ of C_4 such that

$$\rho(\mathbf{g}^k)f(I, x) = f(P_{90}^k I, R_{90}^k x) \tag{4}$$

for $k \in \{0, 1, 2, 3\}$, where P_{90} and R_{90} are the representations from Examples 3.1 and 3.2 that rotate images and coordinates in the ordinary manner.

Both SIFT and Upright SIFT are equivariant. For SIFT, $\rho(\mathbf{g}^k)$ is the identity, so SIFT is invariant. For Upright SIFT, $\rho(\mathbf{g}^k)$ is P_{UPSIFT}^k as explained in Example 4.1.

One aim of this work is to argue and demonstrate that learned keypoint descriptors, which are trained on upright data, will behave more like Upright SIFT than SIFT, *i.e.*, they will be rotation equivariant but not invariant.

Definition 4.3 (Steerability, adapted from [21])**.** A real-valued function $\phi : V \to \mathbb{R}$ is said to be steerable under a representation ρ_{in} of G on V, if there exist D functions (for some D) $\phi_j : V \to \mathbb{R}$ and D functions $\kappa_j : G \to \mathbb{R}$ such that $\phi(\rho_{\text{in}}(g)v) = \sum_{j=1}^{D} \kappa_j(g)\phi_j(v)$.

Note that an equivariant function $f : V \to \mathbb{R}^D$ satisfies in each component f_d that $f_d(\rho_{\text{in}}(g)v) = \sum_{j=1}^{D} \rho(g)_{dj} f_j(v)$, so each component of f is steerable, in the notation of Definition 4.3, $\phi = f_d, \phi_j = f_j, \kappa_j(g) = \rho(g)_{dj}$. This motivates the definition of a *steerer*.

Definition 4.4 (Steerer)**.** Given a function $f : V \to W$ between vector spaces, and a representation ρ_{in} of G on V, a steerer is a representation ρ of G on W that makes f equivariant, *i.e.* such that

$$f(\rho_{\text{in}}(g)v) = \rho(g)f(v). \tag{5}$$

Even if (5) only holds approximately or ρ is only approximately a representation, we will refer to ρ as a steerer.

We will use the verb *steer* for multiplying a feature/description by a steerer; see Figure 2 for the broad idea.

Example 4.3. As explained in Example 4.1, P_{UPSIFT} is a steerer for Upright SIFT under 90° rotations. This has practical consequences. If we want to obtain the Upright SIFT descriptions for an image I and the same image rotated $k \cdot 90°$, we only need to compute the descriptions for the original image. We can obtain the rotated ones by multiplying the descriptions by P_{UPSIFT}^k. That is, we can steer the Upright SIFT descriptions with P_{UPSIFT}.

It is known from representation theory [47] what all possible representations of C_4 are, and hence what all possible steerers for rotation equivariant descriptors are. As this result will be necessary for the remainder of the text, we collect it in a theorem. Similar results are also known for other groups *e.g.* the continuous rotation group $\mathrm{SO}(2)$, which we discuss in the next section.

Theorem 4.1 (Representations of C_4)**.** Let ρ be a representation of C_4 on $\mathbb{R}^D$. Then, there exists an invertible matrix Q and $j_d \in \{0, 1, 2, 3\}$ such that

$$\rho(\mathbf{g}^k) = Q^{-1}\mathrm{diag}(\mathbf{i}^{kj_1}, \mathbf{i}^{kj_2}, \ldots, \mathbf{i}^{kj_D})Q. \tag{6}$$

The diagonal in (6) contains the eigenvalues of $\rho(\mathbf{g}^k)$.

Example 4.4. The Upright SIFT steerer P_{UPSIFT} is diagonalizable with an equal amount of each eigenvalue ± 1, $\pm\mathbf{i}$.

The complex eigenvalues must appear in conjugate pairs as we take $\rho(\mathbf{g})$ to be real-valued. It is then possible to do a change of basis so that each pair $\mathbf{i}, -\mathbf{i}$ on the diagonal in (6) is replaced by a block $\left(\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right)$. In this way, $\rho(\mathbf{g})$ can always be block-diagonalized: $\rho(\mathbf{g}) = Q^{-1}BQ$ where Q and B are real valued and B is block-diagonal with blocks of sizes 1 and 2.

4.1. Representation theory of $\mathrm{SO}(2)$

$\mathrm{SO}(2)$ is a one-parameter Lie group, *i.e.* a continuous group with one degree of freedom α—the rotation angle. A D-dimensional representation of $\mathrm{SO}(2)$ is a map $\varsigma : [0, 2\pi) \to \mathrm{GL}(\mathbb{R}, D)$ such that addition modulo 2π on the input is encoded as matrix multiplication on the output—we will consistently use ς for $\mathrm{SO}(2)$ representations to separate them from C_4 representations ρ (ρ will also be used for representations of general groups). The most familiar is the two-dimensional representation $\varsigma(\alpha) = \left(\begin{smallmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{smallmatrix}\right)$ which rotates 2D coordinates. Similar to the C_4 case in Theorem 4.1, we can write down a general representation for $\mathrm{SO}(2)$ as follows [57].

Theorem 4.2 (Representations of $\mathrm{SO}(2)$)**.** Let ς be a representation of $\mathrm{SO}(2)$ on $\mathbb{R}^D$. Then there exists an invertible Q and $j_d \in \mathbb{Z}$ such that

$$\varsigma(\alpha) = Q^{-1}\mathrm{diag}\left(\mathrm{e}^{\mathbf{i}j_1\alpha}, \mathrm{e}^{\mathbf{i}j_2\alpha}, \ldots, \mathrm{e}^{\mathbf{i}j_D\alpha}\right)Q. \tag{7}$$

The j_d's are the frequencies of the eigenspaces of ς. Complex eigenvalues appear in conjugate pairs so (7) can be rewritten as a block diagonal decomposition $\varsigma(\alpha) = Q^{-1}BQ$ where Q and B are real valued and B is block-diagonal with minimal blocks. The admissible blocks (= real-valued irreducible representations) in B are then the 1×1 block $\begin{pmatrix}1\end{pmatrix}$ and the 2×2 blocks

$$\begin{pmatrix}\cos(j\alpha) & -\sin(j\alpha) \\ \sin(j\alpha) & \cos(j\alpha)\end{pmatrix} \quad \text{for } j \in \mathbb{Z} \setminus \{0\}. \tag{8}$$

We can write $\varsigma(\alpha) = \operatorname{expm}\left(\alpha Q^{-1}\operatorname{diag}(\mathbf{i}j_1, \ldots, \mathbf{i}j_D)Q\right)$ where expm is the matrix exponential. The quantity $\mathrm{d}\varsigma := Q^{-1}\operatorname{diag}(\mathbf{i}j_1, \ldots, \mathbf{i}j_D)Q$ is called a Lie algebra representation of $\mathrm{SO}(2)$, here in its most general form. When training a steerer for $\mathrm{SO}(2)$ it is practical to train a $D \times D$ matrix $\mathrm{d}\varsigma$ and steer using $\varsigma(\alpha) = \operatorname{expm}(\alpha \mathrm{d}\varsigma)$.

4.2. Disentangling description space

When we have a steerer, we get a description space on which rotations act—up to a change of basis—by a block-diagonal matrix $\oplus_{j=1}^{J} B_j$. The description space can then be thought of as being disentangled into different subspaces where rotations act in different ways B_j [14, 60]. We detail what this means for keypoint descriptors in Appendix A.

5. Descriptors and steerers

This work's crucial observation and assumption is that learned descriptors, while not invariant, are approximately equivariant so that they have a steerer. Or, as a weaker assumption, they can be trained to be equivariant. It may seem that this is a strong assumption. However, a seemingly less strong assumption turns out to be equivalent.

Theorem 5.1. [Adapted from Shakerinava et al. [48], Gupta et al. [25]] Assume that we have a function $f : V \to \mathbb{S}^{D-1}$ and a group G with representation ρ_{in} on V such that, for all $v, v' \in V$ and $g \in G$

$$\langle f(\rho_{\text{in}}(g)v), f(\rho_{\text{in}}(g)v')\rangle = \langle f(v), f(v')\rangle. \tag{9}$$

Then there exists an orthogonal representation $\rho(g)$, such that f is equivariant w.r.t. G with representations ρ_{in} and ρ.

We provide a proof in Appendix A. Since we match normalized keypoint descriptions by their cosine similarity, Theorem 5.1 is highly applicable to the image matching problem. If a keypoint descriptor f is perfectly consistent in the matching scores when *simultaneously* rotating the images, then the scalar products in (9) will be equal so that the theorem tells us that f has a steerer ρ. Furthermore, we can expect many local image features to appear in all orientations even over a dataset of upright images, thus encouraging (9) to hold for f trained on large datasets.

5.1. Three settings for investigating steerers

As C_4 is cyclic, all its representations are defined by $\rho(\mathbf{g})$, where $\mathbf{g}$ is the generator of C_4. To find a steerer for a keypoint descriptor under C_4 hence comes down to finding a single matrix $\rho(\mathbf{g})$ that represents rotations by 90° in the description space. Similarly, for $\mathrm{SO}(2)$ we find the single matrix $\mathrm{d}\varsigma$ that defines the representation ς.

We will consider three settings. In each case we will optimize $\rho(\mathbf{g})$ and/or f over the MegaDepth training set [33] with rotation augmentation and maximize

$$p\left(f(P_{90}^{k_1} I_1, R_{90}^{k_1} x_1), \rho(\mathbf{g})^k f(P_{90}^{k_2} I_2, R_{90}^{k_2} x_2)\right) \tag{10}$$

where p is the matching probability (1). The number of rotations k_1 and k_2 for each image are sampled independently during training, and $k = k_1 - k_2 \mod 4$ is the number of rotations that aligns image I_2 to image I_1. Thus, $\rho(\mathbf{g})^k$ aligns the relative rotation between descriptions in I_2 and I_1. We optimize continuous steerers ς analogously to (10).

Setting A: Fixed descriptor, optimized steerer. If a descriptor works equally well for upright images as well as images rotated the same amount from upright, then according to Theorem 5.1, we should expect that there exists a steerer $\rho(\mathbf{g})$ such that (4) holds. To find $\rho(\mathbf{g})$ we optimize it as a single $D \times D$ linear layer by maximizing (10).

Setting B: Joint optimization of descriptor and steerer. The aim is to find a steerer that is as good as possible for the given data. We will see in the experiments, by looking at the evolution of the eigenvalues of $\rho(\mathbf{g})$ during training, that this joint optimization has many local optima and is highly dependent on the initialization of $\rho(\mathbf{g})$. However, looking at the eigenvalues of $\rho(\mathbf{g})$ does give knowledge about which descriptor dimensions are most important, as will be explained in Section 6.5.

Setting C: Fixed steerer, optimized descriptor. To get the most precise control over the rotation behaviour of a descriptor, we can fix the steerer and optimize only the descriptor. This enables us to investigate how much influence the choice of steerer has on the descriptor. For instance, choosing the steerer as the identity leads to a rotation invariant descriptor. We will see in the experiments that this choice leads to suboptimal performance on upright images compared to other steerers.

5.2. Matching with equivariant descriptions

This section presents several approaches to rotation invariant matching using equivariant descriptors. Throughout, we will denote the D-dimensional descriptions of N keypoints in two images I_1, I_2 by $y_1, y_2 \in \mathbb{R}^{D \times N}$ and will assume that we know the C_4-steerer $\rho(\mathbf{g})$ that rotates descriptions 90° or the $\mathrm{SO}(2)$-steerer $\varsigma(\alpha)$ through the Lie algebra generator $\mathrm{d}\varsigma$. For matching we follow DeDoDe [19], as described in Section 3.1. The base similarity used is the cosine

Table 1. **Evaluation on Roto-360 [31].** We report the percentage of correct matches at three thresholds. We use the DeDoDe-SO2 detector with $5,000$ keypoints in the last two rows. Matching strategies are Max Matches for the C_4-descriptor and Max Sim. over C_8 for the $\mathrm{SO}(2)$-descriptor. See Section 6.1 for the shorthands for our models.

Detector	Descriptor	3px	5px	10px
SIFT [35]	SIFT [35]	78	78	79
ORB [45]	ORB [45]	79	85	87
SuperPoint [17]	RELF, single [31]	90	91	93
SuperPoint [17]	RELF, multiple [31]	92	93	94
SuperPoint [17]	C4-B (ours)	82	82	83
SuperPoint [17]	SO2-Spread-B (ours)	**96**	**97**	97
DeDoDe [19]	C4-B (ours)	82	84	86
DeDoDe [19]	SO2-Spread-B (ours)	95	**97**	**98**

similarity, so we compute $y_1^T y_2$ for normalized descriptions to get an $N \times N$ matrix of pairwise scores on which dual softmax (1) is applied. Matches are mutual most similar descriptions with similarity above 0.01.

Max matches over steered descriptions. The first way of obtaining invariant matches is to match y_1 with $\rho(\mathbf{g})^k y_2$ for $k = 0, 1, 2, 3$ and keep the matches from the k that has the most matches. This is similar to matching the image I_1 with four different rotations of I_2 but alleviates the need for rerunning the descriptor network for each rotation.

Max similarity over steered descriptions. A computationally cheaper version is to select the matching matrix not as $y_1^T y_2$ but as $\max_k y_1^T \rho(\mathbf{g})^k y_2$, where the $\max$ is elementwise over the matrix.

$\mathrm{SO}(2)$-steerers. To apply the above matching strategies to $\mathrm{SO}(2)$-steerers $\varsigma(\alpha) = \operatorname{expm}(\alpha \mathrm{d}\varsigma)$ we discretize ς. A C_ℓ-steerer is obtained through $\rho(\mathbf{g}_\ell) = \operatorname{expm}\left(\frac{2\pi}{\ell}\mathrm{d}\varsigma\right)$, where $\mathbf{g}_\ell$ generates C_ℓ. We will use C_8 in the experiments.

Procrustes matcher. If all eigenvalues of $\rho(\mathbf{g})$ are $\pm\mathbf{i}$, the steerer can be block-diagonalized with only the block $\left(\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right)$[3]. The descriptions consist of $D/2$ two-dimensional quantities that all rotate with the same frequency as the image. We will refer to them as frequency 1 descriptions and view them reshaped as $y \in \mathbb{R}^{2\times(D/2)\times N}$. A 2D rotation matrix acts on these descriptions from the left when the image rotates, and we can find the optimal rotation matrix $R_{m,n}$ that aligns each pair $y_{1,m}, y_{2,n} \in \mathbb{R}^{2\times(D/2)}$ by solving the Procrustes problem via SVD. The matching matrix is obtained by computing $\langle R_{m,n} y_{1,m}, y_{2,n} \rangle$ for each pair. $R_{m,n}$ gives the relative rotation between each pair of keypoints, which can be useful *e.g.* for minimal relative pose solvers [4, 6] or for outlier filtering [13]. We leave exploring this per-correspondence geometry to future work.

Table 2. **Evaluation on AIMS [49].** We report the average precision (AP) in percent on different splits of AIMS: “North Up” (N. Up) contains images with small rotations, “All Others” (A. O.) contains images with larger rotations and “All” contains all images. We use the DeDoDe-SO2 detector and $10,000$ keypoints throughout. See Section 6.1 for the shorthands for our models.

Method	N. Up	A. O.	All
SE2-LoFTR [9]	58	51	52
C4-B, Max Matches (ours)	52	51	51
SO2-Spread-B, Max Sim. C8 (ours)	60	57	58
SO2-Freq1-B, Procrustes (ours)	**64**	**59**	**60**

6. Experiments

We train and evaluate a variety of descriptors and steerers. Experimental details are covered in Appendix C. We provide comparisons to TTA in performance and runtime in Appendix B, as well as experiments with more matching strategies and an explicit experiment to test the connection between Theorem 5.1 and rotation equivariance.

We start by reporting results on two public benchmarks for rotation invariant image matching. Then, we will present ablation results for the MegaDepth benchmark, both for the standard version with upright images and a version where we have rotated the input images.

6.1. Models considered

Our base models are the DeDoDe-B and DeDoDe-G descriptors introduced in [19]. These are both $D = 256$ dimensional descriptors. The focus will be on the smaller model DeDoDe-B, as this gives us the chance to do large-scale ablations. We train all models on MegaDepth [33]. To obtain rotation-consistent detections, we retrain two versions of the DeDoDe-detector, with data augmentation over C_4 and $\mathrm{SO}(2)$ respectively, denoted DeDoDe-$\{$C4, SO2$\}$.

For Setting B (Section 5.1), we will see that the initialization of the steerer matters. Similarly, for Setting C, we can fix the steerer with different eigenvalue structures. Here, we introduce shorthand, which is used in the result tables. We will refer to the case of all eigenvalues 1 as *Inv* for invariant. This case corresponds to the ordinary notion of data augmentation, where the descriptions for rotated images should be the same as for non-rotated images. The case when all eigenvalues are $\pm\mathbf{i}$ is denoted *Freq1* for frequency 1 as explained in Section 5.2. For C_4-steerers, the case with an equal distribution of all eigenvalues $\pm 1, \pm\mathbf{i}$ will be denoted *Perm*, as this is the eigenvalue signature of a cyclic permutation of order 4. For $\mathrm{SO}(2)$-steerers, the case with an equal distribution of eigenvalues $0, \pm\mathbf{i}, \pm 2\mathbf{i}, \ldots \pm 6\mathbf{i}$ will be denoted *Spread* (the cutoff 6 was arbitrarily chosen). The *Perm* and *Spread* steerers correspond to a broad range of frequencies in the description space. When none of the above labels (*Inv*, *Freq1*, *Perm* or

[3] This also holds for $\mathrm{SO}(2)$ steerers, referring to eigenvalues and blocks of the Lie algebra generator $\mathrm{d}\varsigma$.

Table 3. **Evaluation on MegaDepth [33, 50]**. The first section shows Setting A where we only optimize the steerer, the second section shows Setting B where we jointly optimize the descriptor and steerer and the third section shows the Setting C where we predefine the steerer and optimize only the descriptor. For MegaDepth-1500 we always use dual softmax matcher to evaluate the descriptors on upright images. We use $20,000$ keypoints throughout. The best values for **B**- and **G**-models are highlighed in each column. See Section 6.1 for shorthand explanations for our models. A larger version of this table with more methods is available in Appendix B.

Detector	Descriptor		MegaDepth-1500			MegaDepth-C4			MegaDepth-SO2		
DeDoDe	DeDoDe	Matching strategy	AUC @ 5°	10°	20°	5°	10°	20°	5°	10°	20°
Original	B	Dual softmax	49	65	77	12	17	20	12	16	20
Original	B	Max matches C4-steered	-"-	-"-	-"-	43	60	73	30	44	56
SO2	B	Max matches C8-steered	50	66	78	40	57	70	34	51	65
Original	G	Dual softmax	**52**	**69**	**81**	13	17	21	16	22	28
Original	G	Max matches C4-steered	-"-	-"-	-"-	31	45	57	26	39	50
C4	C4-B	Max matches C4-steered	**51**	**67**	**79**	**50**	**67**	**79**	39	55	68
SO2	SO2-B	Max matches C8-steered	47	63	76	47	63	76	44	61	74
SO2	SO2-Spread-B	Max matches C8-steered	50	66	**79**	49	66	78	**46**	**63**	**76**
SO2	SO2-Spread-B	Max similarity C8-steered	49	66	78	47	64	77	43	61	74
C4	C4-Inv-B	Dual softmax	48	64	76	47	63	76	39	55	69
C4	C4-Perm-B	Max matches C4-steered	50	**67**	**79**	**50**	66	**79**	39	54	67
SO2	SO2-Inv-B	Dual softmax	46	62	75	45	61	74	43	60	73
SO2	SO2-Freq1-B	Max matches C8-steered	47	64	77	47	64	76	45	62	75
SO2	SO2-Freq1-B	Procrustes	47	64	76	46	62	75	45	61	74
C4	C4-Perm-G	Max matches C4-steered	**52**	**69**	**81**	**53**	**69**	**82**	**44**	**61**	**74**

Spread) is attached to a descriptor and steerer trained jointly in Setting B, then we initialize the steerer with values uniformly in $(-D^{-1/2}, D^{-1/2})$[4], the eigenvalues are then approximately uniformly distributed in the disk with radius $3^{-1/2}$ [51]. When a descriptor is trained with $k \cdot 90°$ rotations, we append *C4* to its name and when trained with continuous augmentations, we append *SO2* to its name.

6.2. Roto-360

We evaluate on the Roto-360 benchmark [31], which consists of ten image pairs from HPatches [3] where the second image in each image pair is rotated by all multiplies of 10° to obtain 360 image pairs in total. We report the average precision of the obtained matches and compare it to the current state-of-the-art RELF [31]. The results are shown in Table 1. We see that we outperform RELF when using methods trained for continuous rotations. Our matching runs around three times faster than RELF on Roto-360.

6.3. AIMS

The Astronaut Image Matching Subset (AIMS) [49] consists of images taken by astronauts from the ISS and satellite images covering the broad regions that the astronaut images could depict. The task consists of finding the pairs of astronaut images and satellite images that show the same locations on Earth. Pairs are found by setting a threshold for the number of matches between images after homography estimation with RANSAC.

The relative rotations of the astronaut and satellite images are unknown, making the task suitable for rotation-invariant matchers. Indeed, in [49], the best performing method is the rotation invariant SE2-LoFTR [9], which we compare to. The AIMS can be split into “North Up” astronaut images, consisting of images with small rotations (between 0° and 90°) and “All Others”, consisting of images with large rotations. This split further enables the evaluation of rotation invariant matchers. We report the average precision over the whole dataset, as opposed to the approach in [49], where the score is computed over at most 100 true negatives per astronaut image. Results are shown in Table 2. Further, we plot precision-recall curves in Appendix B. We generally outperform SE2-LoFTR, particularly on the heavily rotated images in “All Others”.

6.4. MegaDepth-1500

We evaluate on a held-out part of MegaDepth (MegaDepth-1500 following [50]). Here, the task is to take two input images and output the relative pose between the cameras. The performance is measured by the AUC of the pose error. Additionally, we create two versions of MegaDepth with rotated images to evaluate the rotational robustness of our models. For MegaDepth-C4, the second image in every image pair is rotated $(i \mod 4) \cdot 90°$ where i is the index of the image pair. We visualize a pair in MegaDepth-C4 in Appendix B, illustrating the improvement from DeDoDe-B to DeDoDe-B with a steerer optimized in Setting A. For MegaDepth-SO2, the second image in every image pair is

[4] This is the standard initialization of a linear layer in Pytorch [26, 40].

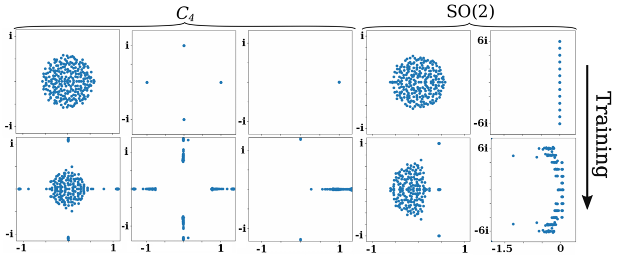

Figure 4. **Training evolution of eigenvalue distributions of steerers.** We plot the eigenvalue distribution of C_4-steerers $\rho(\mathbf{g})$ (first three columns) and Lie algebra generators $\mathrm{d}\varsigma$ for $\mathrm{SO}(2)$-steerers (last two columns) in the complex plane, with different initializations when trained jointly with a descriptor. The top row depicts the eigenvalues at the start, and the bottom row at the end of training. There are $D = 256$ eigenvalues in every plot—many congregate at the "admissible" eigenvalues as described in Section 4—but some do not, see the discussion in Section 6.5. These visualizations highlight the initialization sensitivity of the steerer. We show gif movies of the training evolution at this https url.

instead rotated $(i \mod 36) \cdot 10°$, thus requiring robustness under continuous rotations.

The results are presented in Table 3; for more methods, see Appendix B. We summarize the main takeaways:

1. It is possible to find steerers for the original DeDoDe models (*e.g.* the second row of the table), even though they were not trained with any rotation augmentation.
2. The trained C_4 steerers perform very well as their scores on MegaDepth-1500 and MegaDepth-C4 are the same.
3. Training DeDoDe-B jointly with a C_4 steerer (C4-B) or with a fixed steerer (C4-Perm-B) improves results on upright images—this can be attributed to the fact that training with a steerer enables using rotation augmentation.
4. The right equivariance for the task at hand is crucial—$\mathrm{SO}(2)$-steerers outperform others on MegaDepth-SO2.
5. The eigenvalue distribution of the steerer is important—invariant models are worse than others, and SO2-B and SO2-Freq1-B are worse than SO2-Spread-B.
6. DeDoDe-G can be made equivariant (C4-Perm-G), even though it has a frozen DINOv2 [39] ViT backbone.

6.5. Training dynamics of steerer eigenvalues

This section aims to demonstrate that joint optimization of the steerer and descriptor does not necessarily lead to a good eigenvalue structure for the steerer. We plot the evolution of the eigenvalues of the steerer over the training epochs in Figure 4. For C_4-steerers we plot the eigenvalues of $\rho(\mathbf{g})$ itself, while for $\mathrm{SO}(2)$-steerers we plot the eigenvalues λ_d of the Lie algebra generator $\mathrm{d}\varsigma$, so that the eigenvalues of the steerer $\varsigma(\alpha)$ are $\mathrm{e}^{\alpha\lambda_d}$. It is clear from Figure 4 that the initialization of the steerer influences the final distribution of eigenvalues a lot and we saw in Table 3 that the eigenvalue distribution of the steerer matters for performance. Thus, we think it is an important direction for future work to figure out how to get around this initialization sensitivity. The choice of eigenvalue structure is related to the problem of specifying which group representations to use in the layers of equivariant neural networks in general.

As a side effect of plotting the eigenvalues, we find that some of the steerer's eigenvalues have much lower absolute values than others[5]. The steerer is applied to descriptions before they are normalized, so the absolute value of the maximum eigenvalue is unimportant, but the relative size of the eigenvalues tells us something about feature importance. Eigenvectors with small eigenvalues cannot be too important for matching, since they will be relatively downscaled when applying the steerer in the optimization of (10). Indeed, small eigenvalues seem to correspond to unimportant dimensions of the descriptor—we maintain matching performance when projecting the descriptions to the span of the eigenvectors with large eigenvalues. This is related to PCA for dimensionality reduction, which has successfully been used for classical keypoint descriptors [27].

7. Conclusion

We developed a new framework for rotation equivariant keypoint descriptors using steerers—linear maps that encode image rotations in description space. After outlining the general theory of steerers using representation theory, we designed a large set of experiments with steerers in three settings: (A) optimizing a steerer for a fixed descriptor, (B) optimizing a steerer and a descriptor jointly and (C) optimizing a descriptor for a fixed steerer. Our best models obtained new state-of-the-art results on the rotation invariant matching benchmarks Roto-360 and AIMS.

Acknowledgements

We thank Alex Stoken for help with the evaluation on AIMS. This work was supported by the Wallenberg Artificial Intelligence, Autonomous Systems and Software Program (WASP), funded by the Knut and Alice Wallenberg Foundation and by the strategic research environment ELLIIT, funded by the Swedish government. The computational resources were provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS) at C3SE, partially funded by the Swedish Research Council through grant agreement no. 2022-06725, and by the Berzelius resource, provided by the Knut and Alice Wallenberg Foundation at the National Supercomputer Centre.

[5] The absolute values of the eigenvalues of a $\mathrm{SO}(2)$ steerer $\varsigma(\alpha)$ are $\mathrm{e}^{\alpha \mathrm{Re}(\lambda_d)}$ where λ_d are the eigenvalues of $\mathrm{d}\varsigma$ that are plotted in the two rightmost columns of Figure 4. Therefore, a lower real value of λ_d means a lower absolute value of the eigenvalue of the steerer.

References

[1] Georges Baatz, Kevin Köser, David Chen, Radek Grzeszczuk, and Marc Pollefeys. Handling urban location recognition as a 2d homothetic problem. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part VI 11*, pages 266–279. Springer, 2010. 3

[2] Piyush Bagad, Floor Eijkelboom, Mark Fokkema, Danilo de Goede, Paul Hilders, and Miltiadis Kofinas. C-3po: Towards rotation equivariant feature detection and description. In *European Conference on Computer Vision*, pages 694–705. Springer, 2022. 2

[3] 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. 7

[4] Daniel Barath and Zuzana Kukelova. Relative pose from sift features. In *European Conference on Computer Vision*, pages 454–469. Springer, 2022. 6

[5] Daniel Barath, Jana Noskova, Maksym Ivashechkin, and Jiri Matas. Magsac++, a fast, reliable and accurate robust estimator. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1304–1312, 2020. 6

[6] Daniel Barath, Michal Polic, Wolfgang Förstner, Torsten Sattler, Tomas Pajdla, and Zuzana Kukelova. Making affine correspondences work in camera geometry computation. In *European Conference on Computer Vision*, pages 723–740. Springer, 2020. 6

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

[8] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346–359, 2008. Similarity Matching in Computer Vision and Multimedia. 2

[9] Georg Bökman and Fredrik Kahl. A case for using rotation invariant features in state of the art feature matchers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5110–5119, 2022. 2, 6, 7

[10] Georg Bökman and Fredrik Kahl. Investigating how ReLU-networks encode symmetries. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 2

[11] Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021. 2

[12] Robert-Jan Bruintjes, Tomasz Motyka, and Jan van Gemert. What affects learned equivariance in deep image recognition models? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 4838–4846, 2023. 2

[13] Luca Cavalli, Viktor Larsson, Martin Ralf Oswald, Torsten Sattler, and Marc Pollefeys. Adalam: Revisiting handcrafted outlier detection. *arXiv preprint arXiv:2006.04250*, 2020. 6

[14] Taco Cohen and Max Welling. Learning the irreducible representations of commutative lie groups. In *Proceedings of the 31st International Conference on Machine Learning*, pages 1755–1763, Bejing, China, 2014. PMLR. 5, 1

[15] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pages 2990–2999. PMLR, 2016. 2

[16] Taco S Cohen and Max Welling. Transformation properties of learned visual representations. *ICML 2015 (arXiv:1412.7659)*, 2014. 2, 1

[17] 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. 1, 6

[18] Tom Edixhoven, Attila Lengyel, and Jan C van Gemert. Using and abusing equivariance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 119–128, 2023. 2

[19] Johan Edstedt, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. DeDoDe: Detect, Don't Describe – Describe, Don't Detect for Local Feature Matching. In *2024 International Conference on 3D Vision (3DV)*. IEEE, 2024. 1, 3, 5, 6, 7, 8

[20] Marc Finzi, Max Welling, and Andrew Gordon Wilson. A practical method for constructing equivariant multilayer perceptrons for arbitrary matrix groups. In *International conference on machine learning*, pages 3318–3328. PMLR, 2021. 2

[21] William T Freeman, Edward H Adelson, et al. The design and use of steerable filters. *IEEE Transactions on Pattern analysis and machine intelligence*, 13(9):891–906, 1991. 4

[22] Jan E. Gerken, Jimmy Aronsson, Oscar Carlsson, Hampus Linander, Fredrik Ohlsson, Christoffer Petersson, and Daniel Persson. Geometric deep learning and equivariant neural networks. 56(12):14605–14662. 2

[23] Pierre Gleize, Weiyao Wang, and Matt Feiszli. SiLK: Simple Learned Keypoints. In *ICCV*, 2023. 1

[24] Nate Gruver, Marc Anton Finzi, Micah Goldblum, and Andrew Gordon Wilson. The lie derivative for measuring learned equivariance. In *The Eleventh International Conference on Learning Representations*, 2023. 2

[25] Sharut Gupta, Joshua Robinson, Derek Lim, Soledad Villar, and Stefanie Jegelka. Structuring representation geometry with rotationally equivariant contrastive learning. *arXiv preprint arXiv:2306.13924*, 2023. 2, 5, 1

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015. 7

[27] Yan Ke and Rahul Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, pages II–II. IEEE, 2004. 8

[28] Masanori Koyama, Kenji Fukumizu, Kohei Hayashi, and Takeru Miyato. Neural fourier transform: A general approach to equivariant representation learning. *arXiv preprint arXiv:2305.18484*, 2023. 2, 1

[29] Jongmin Lee, Yoonwoo Jeong, and Minsu Cho. Self-supervised learning of image scale and orientation. In *31st British Machine Vision Conference 2021, BMVC 2021, Virtual Event, UK*. BMVA Press, 2021. 2

[30] Jongmin Lee, Byungjin Kim, and Minsu Cho. Self-supervised equivariant learning for oriented keypoint detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4847–4857, 2022. 2

[31] Jongmin Lee, Byungjin Kim, Seungwook Kim, and Minsu Cho. Learning rotation-equivariant features for visual correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21887–21897, 2023. 2, 6, 7

[32] Karel Lenc and Andrea Vedaldi. Understanding image representations by measuring their equivariance and equivalence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2

[33] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2041–2050, 2018. 2, 5, 6, 7

[34] Yuan Liu, Zehong Shen, Zhixuan Lin, Sida Peng, Hujun Bao, and Xiaowei Zhou. Gift: Learning transformation-invariant dense visual descriptors via group cnns. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[35] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 1, 2, 3, 6

[36] Giovanni Luca Marchetti, Gustaf Tegnér, Anastasiia Varava, and Danica Kragic. Equivariant representation learning via class-pose decomposition. In *International Conference on Artificial Intelligence and Statistics*, pages 4745–4756. PMLR, 2023. 2, 1

[37] 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. 2

[38] Yuki Ono, Eduard Trulls, Pascal Fua, and Kwang Moo Yi. LF-Net: Learning local features from images. *Advances in neural information processing systems*, 31, 2018. 2

[39] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *arXiv:2304.07193*, 2023. 8

[40] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 7

[41] Rémi Pautrat, Viktor Larsson, Martin R. Oswald, and Marc Pollefeys. Online invariance selection for local feature descriptors. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 1

[42] Nicolas Pielawski, Elisabeth Wetzer, Johan Öfverstedt, Jiahao Lu, Carolina Wählby, Joakim Lindblad, and Nataša Sladoje. CoMIR: Contrastive multimodal image representation for registration. In *Advances in Neural Information Processing Systems*, pages 18433–18444. Curran Associates, Inc., 2020. 1

[43] 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. 1

[44] I. Rocco, M. Cimpoi, R. Arandjelović, A. Torii, T. Pajdla, and J. Sivic. Neighbourhood consensus networks. In *Proceedings of the 32nd Conference on Neural Information Processing Systems*, 2018. 3

[45] 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. 2, 6

[46] Emanuele Santellani, Christian Sormann, Mattia Rossi, Andreas Kuhn, and Friedrich Fraundorfer. S-trek: Sequential translation and rotation equivariant keypoints for local feature extraction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9728–9737, 2023. 2

[47] Jean-Pierre Serre. *Linear Representations of Finite Groups*. Springer, 1977. 4

[48] Mehran Shakerinava, Arnab Kumar Mondal, and Siamak Ravanbakhsh. Structuring representations using group invariants. In *Advances in Neural Information Processing Systems*, pages 34162–34174. Curran Associates, Inc., 2022. 2, 5, 1

[49] Alex Stoken and Kenton Fisher. Find my astronaut photo: Automated localization and georectification of astronaut photography. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 6196–6205, 2023. 1, 2, 6, 7, 4, 5

[50] 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*, pages 8922–8931, 2021. 2, 3, 7

[51] Terence Tao, Van Vu, and Manjunath Krishnapur. Random matrices: Universality of ESDs and the circular law. *The Annals of Probability*, 38(5):2023 – 2065, 2010. 7

[52] 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. 2

[53] 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. 2

[54] 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*, 33:7401–7412, 2020. 2

[55] Michal J. Tyszkiewicz, Pascal Fua, and Eduard Trulls. DISK: learning local features with policy gradient. In *NeurIPS*, 2020. 1, 3, 7

[56] Maurice Weiler and Gabriele Cesa. General e (2)-equivariant steerable cnns. *Advances in neural information processing systems*, 32, 2019. 2

[57] Peter Woit. *Quantum Theory, Groups and Representations*. Springer International Publishing, 2017. 4

[58] Jeffrey Wood and John Shawe-Taylor. Representation theory and invariant neural networks. *Discrete Applied Mathematics*, 69(1):33–60, 1996. 2

[59] 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 (CVPR)*, 2017. 2

[60] Daniel E. Worrall, Stephan J. Garbin, Daniyar Turmukhambetov, and Gabriel J. Brostow. Interpretable transformations with encoder-decoder networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 2, 5, 1

[61] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. Lift: Learned invariant feature transform. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI 14*, pages 467–483. Springer, 2016. 2