DeMatch: Deep Decomposition of Motion Field for Two-View Correspondence Learning

Shihua Zhang, Zizhuo Li, Yuan Gao, Jiayi Ma*
Electronic Information School, Wuhan University, Wuhan 430072, China
suhzhang001@gmail.com, zizhuo_li@whu.edu.cn, {ethan.y.gao, jyma2010}@gmail.com

Abstract

Two-view correspondence learning has recently focused on considering the coherence and smoothness of the motion field between an image pair. Dominant schemes include controlling the complexity of the field function with regularization or smoothing the field with local filters, but the former suffers from heavy computational burden, and the latter fails to accommodate discontinuities in the case of large scene disparities. In this paper, inspired by Fourier expansion, we propose a novel network called DeMatch, which decomposes the motion field to retain its main "low-frequency" and smooth part. This achieves implicit regularization with lower computational cost and generates piecewise smoothness naturally. Specifically, we first decompose the rough motion field that is contaminated by false matches into several different sub-fields, which are highly smooth and contain the main energy of the original field. Then, with these smooth sub-fields, we recover a cleaner motion field from which correct motion vectors are subsequently derived. We also design a special masked decomposition strategy to further mitigate the negative influence of false matches. All the mentioned processes are finally implemented in a discrete and learnable manner, avoiding the difficulty of calculating real dense fields. Extensive experiments reveal that DeMatch outperforms state-of-the-art methods in multiple tasks and shows promising low computational usage and piecewise smoothness property. The code and trained models are publicly available at https://github.com/SuhZhang/DeMatch.

1. Introduction

Finding two-view correspondences that indicate the same scene points from different perspectives is a fundamental problem in computer vision [21]. The geometry relationship between two-view images is estimated after establishing sparse correspondences, serving as a critical prerequisite for many problems such as panoramic stitching [4], structure from motion [30], and simultaneous location and mapping [23]. A typical matching pipeline starts with identifying keypoints and constructing local descriptions, this allows the generation of a putative match set by assessing the similarity of descriptions [22]. While SIFT [18] is one of the most widely used handcrafted descriptors, learning-based ones have been investigated [7, 24, 34]. However, due to the limited discriminative ability of descriptors, numerous false matches (outliers) are found in the putative set because of wide variations in viewpoint or excessive repetitive patterns. Thus outlier rejection is applied to determine the true matches (inliers), which is the focus of this paper.

Outlier rejection that builds reliable two-view correspondences has been studied at an early age. Most of the classical methods try seeking deformation consistency, *i.e.* motion field consensus and smoothness, to identify inliers. Some of them apply global consensus directly [9, 19],

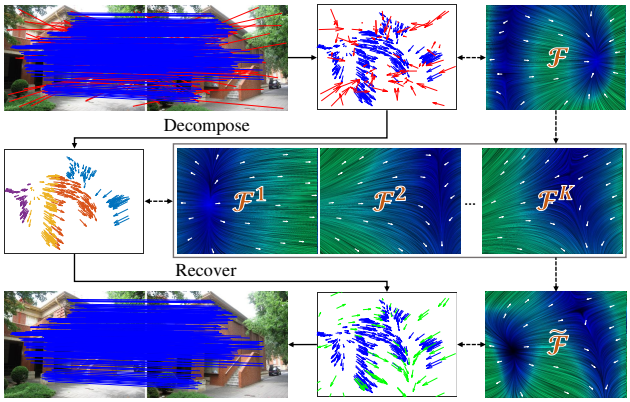


Figure 1. Deep decomposition framework. Lines colored blue are inliers, red are outliers, green are corrected outliers, except for the motion clusters with different colors. The solid lines indicate the workflow of DeMatch, dashed lines correspond to the changes of the motion field, bidirectional arrows represent the conversions between sparse motion vectors and the dense motion field. $\mathcal{F}$ means the original rough motion field, $\mathcal{F}^k$ means the k-th sub-field, and $\widetilde{\mathcal{F}}$ means the recovered cleaner motion field.

*Corresponding author

while others relax the global constraint to local consistency [3, 20]. However, traditional methods often fail in real scenes where the potential motion field is heavily contaminated by heavy outliers, so learning-based approaches are emerging. PointCN [41] marks correspondences with inliers and outliers, modeling outlier rejection as a classification problem. Then, powerful information embedding modules are designed to retrieve and aggregate global and local context [6, 16, 42, 44]. Although these directly classifying methods have shown competent ability, the prior coherence and smoothness of the motion field are ignored. To this end, LMCNet [17] considers global coherence by solving a Laplacian regularization term with deep features, and ConvMatch [43] smooths motion field with convolutional neural network (CNN) as local filters to achieve local consistency. By means of the inherent coherence attribute, the performance boosts, but there are still flaws in such motion coherence-based approaches. Tackling the regularization term is arithmetic intensive in LMCNet. And the discontinuities in the motion field caused by large scene disparities are easily over-smoothed by CNN blocks in ConvMatch. Hence, it is imperative to find a method based on motion coherence that addresses the mentioned issues: **(i)** *Eliminate explicitly using of regularization term to avert high computational usage*. **(ii)** *Consider the discontinuities in the motion field and handle the problem of piecewise smoothness.*

1.1. Motivation

It is well-known that any complex function or field with limited energy can be decomposed into simpler sub-functions according to a basis[1]. For example, in Fourier expansion [11], a function can be decomposed into an infinite series of trigonometric functions, while the standard sine functions with different frequencies act as the basis. By selecting the sub-functions on merely a finite low-frequency basis, we can control the complexity of the original function. And the new function recovered by these sub-functions is smoother while retaining the main energy of the original one. This constrains the function to a low-rank subspace, acting as an implicit regularization [9]. Similarly, if we can decompose the rough motion field contaminated by outliers into several highly smooth sub-fields just like decomposing the complex function, a cleaner motion field can be recovered from these sub-fields, which have simple functional representations and can be deemed as the "low-frequency" part relying on a low-frequency basis or a few low-frequency factors[2] [9, 14]. With the decomposition, the potential motion field can be constrained in a low-rank subspace analogously so that the smoothness property is guaranteed and implicit regularization is achieved [38, 39].

However, decomposing the potential dense motion field with only sparse motion vectors as input is hard. Luckily, as shown in Figure 1, motion vectors can be regarded as a discretized representation of motion field [43]. Moreover, the motion vectors that are close in position or belong to the same object or plane often exhibit strong consistency, they share the same motion pattern and are more likely to yield a smooth sub-field [14]. Thus, grouping motion vectors that belong to a particular pattern into the same cluster is equivalent to decomposing the original contaminated motion field $\mathcal{F}$ into several extremely smooth sub-fields $\{\mathcal{F}^k|k=1,\dots,K\}$, and the motion patterns acting as the clustering rules can be regarded as the basis.

1.2. Consideration and Contribution

Based on the above analysis, we propose DeMatch for two-view correspondence learning that incorporates prior knowledge of smoothness and coherence. It conducts a deep decomposition of the motion field as Figure 1. Specifically, to decompose the potential dense motion field $\mathcal{F}$ in a discrete manner, we identify a set of learnable motion patterns as a basis and cluster the putative motion vectors by these patterns. Hence, a few highly smooth sub-fields are generated, $\{\mathcal{F}^k|k=1,\dots,K\}$, and they contain the main energy of $\mathcal{F}$. These sub-fields are almost entirely supported by inliers, allowing us to recover a cleaner motion field $\widetilde{\mathcal{F}}$ that regularizes $\mathcal{F}$ implicitly. By comparing the motion vectors in $\mathcal{F}$ and $\widetilde{\mathcal{F}}$, we can distinguish inliers and outliers. Because these extremely smooth sub-fields, from which the purer motion field $\widetilde{\mathcal{F}}$ is estimated, respectively correspond to unique motion patterns, DeMatch spontaneously achieves piecewise smoothness. In this way, DeMatch effectively addresses both of the issues mentioned above. Additionally, in order to further mitigate the influence of outliers in the decomposition process, we design a masked decomposition strategy that removes known distinct outliers.

In summary, our main contributions are as follows: **(i)** Rather than explicit regularization that suffers from high computational usage or direct smoothing that ignores the discontinuities caused by multiple motion flows, we decompose the field into finite highly smooth sub-fields which belong to different motion patterns. Then the regularization is implicitly achieved and the piecewise smoothness is naturally fulfilled. **(ii)** We implement the decomposition of the motion field in a discrete and learnable manner. It is the first time learning two-view correspondences with a deep decomposition of the motion field. We also devise a masked decomposition strategy to further mitigate the influence of outliers. **(iii)** We design a network DeMatch leveraging deep decomposition of the motion field. We evaluate its effectiveness on multiple tasks, which outperforms the state-of-the-arts. We also further demonstrate the effect of

[1] In mathematics, a set $\boldsymbol{B}$ of vectors $\{\boldsymbol{b}_i\}$ in a vector space $\boldsymbol{V}$ is a basis.

[2] Strictly, a basis must satisfy linear independence condition. However, the redundant factors do not affect the representation of the smoothness subspace. Therefore, we refer to these factors as "basis" in the following.

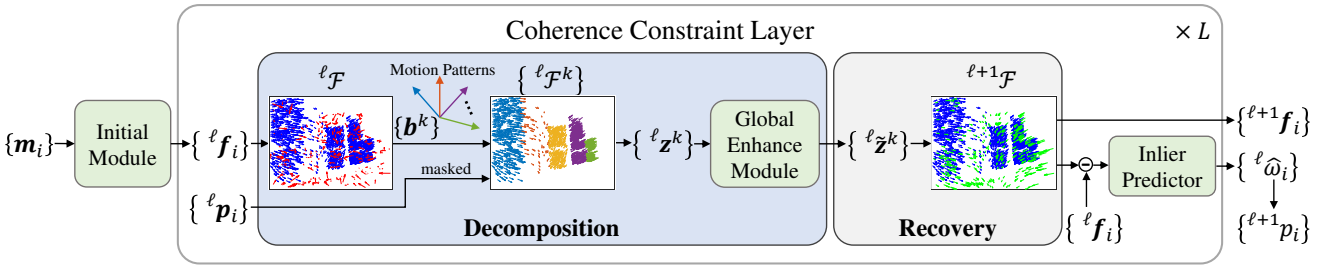


Figure 2. Architecture of DeMatch. Coherence Constraint Layer includes Decomposition and Recovery blocks. After mapping motion vectors into deep space, $\{{}^{\ell}\boldsymbol{f}_i\}$ that describe motion field ${}^{\ell}\mathcal{F}$ are clustered by several motion patterns $\{\boldsymbol{b}^k\}$, thus ${}^{\ell}\mathcal{F}$ is decomposed into finite sub-fields $\{{}^{\ell}\mathcal{F}^k\}$, which are represented by $\{{}^{\ell}\boldsymbol{z}^k\}$. The masked strategy is applied during the decomposition with inlier probability $\{{}^{\ell}p_i\}$ from the former layer. Then global enhancement is conducted, and a cleaner motion field ${}^{\ell+1}\mathcal{F}$ described by $\{{}^{\ell+1}\boldsymbol{f}_i\}$ is recovered with enhanced representations $\{{}^{\ell}\tilde{\boldsymbol{z}}^k\}$. Finally, the output logits $\{{}^{\ell}\hat{\omega}_i\}$ are used to calculate loss and probability $\{{}^{\ell+1}p_i\}$.

all the components and DeMatch's promising properties.

2. Method

The crucial innovation of DeMatch lies in the deep decomposition of the motion field to incorporate the prior knowledge of smoothness and coherence, where complex explicit regularization can be realized implicitly and piecewise smoothness property can be achieved naturally. We first define several learnable tokens as motion patterns and assign putative motion vectors into different clusters by these patterns. Each cluster can generate a highly smooth sub-field so that the original motion field is decomposed, and the motion patterns are regarded as a basis. The architecture of DeMatch is shown in Figure 2. Specifically, after the initial module that upgrades motion vectors into high dimensional space, we decompose the original motion field into several sub-fields depending on the learnable motion patterns as the basis, where the masked decomposition strategy is executed and global information is enhanced. Then we recover a cleaner motion field with these highly smooth sub-fields so that we derive correct motion vectors. Finally, the inlier/outlier classification results are predicted according to the ultimate motion vectors. DeMatch is stacked L times totally with the operations of Decomposition and Recovery. Next, we present the new approach in detail.

2.1. Initialization of Motion Vector

The sparse motion vectors are often used as the input of motion coherence-based outlier rejection methods [17, 19]. Once given N putative correspondences $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)|i = 1, \dots, N,\ \boldsymbol{x}_i \in \mathbb{R}^2,\ \boldsymbol{y}_i \in \mathbb{R}^2\}$, the putative motion vectors are calculated by $\{\boldsymbol{m}_i = (\boldsymbol{x}_i, \boldsymbol{d}_i)|i = 1, \dots, N,\ \boldsymbol{m}_i \in \mathbb{R}^4\}$, where $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ are the coordinates of two corresponding keypoints, and $\boldsymbol{d}_i = \boldsymbol{y}_i - \boldsymbol{x}_i$ is the displacement. Furthermore, for learning-based methods particularly, high dimensional motion vectors are needed to learn better deep features [17, 43]. Hence we represent the motion vectors in a high dimensional space at the very beginning of DeMatch (*i.e.* layer 0) as an initialization:

$$ {}^{0}\boldsymbol{f}_i = \mathcal{E}(\boldsymbol{m}_i), \quad i = 1, \dots, N. \tag{1} $$

$\mathcal{E}(\cdot)$ tries to upgrade the dimension of $\boldsymbol{m}_i \in \mathbb{R}^4$ to ${}^{0}\boldsymbol{f}_i \in \mathbb{R}^C$ by conducting positional embedding [35] ($C = 128$ as default). Details of the initial module are shown in the Supplementary Material (*S.M.*).

2.2. Decomposition of Motion Field

As the main component of DeMatch, the decomposition block starts with defining a few learnable tokens of motion patterns as the basis. Motion vectors can be clustered according to different patterns, then the original motion field is decomposed to several highly smooth sub-fields formed by these clusters, where motion coherence is guaranteed. Additionally, we also introduce masked decomposition strategy and global information enhancement which contribute a lot to the performance.

Main Process of Decomposition. A dense motion field $\mathcal{F} : \mathbb{X} \to \mathbb{D}, \mathcal{F} \in \mathbb{F}, \boldsymbol{x}_i \in \mathbb{X}, \boldsymbol{d}_i \in \mathbb{D}$ is objectively existent in two-view images, which describes the correspondence of every point in the image space, where $\mathbb{F}$ is the functional space, $\mathbb{X} \subseteq \mathbb{R}^2$ is the image space and $\mathbb{D} \subseteq \mathbb{R}^2$ is the displacement space. However, $\mathcal{F}$ is heavily contaminated by a large number of outliers, whose effects are modeled as additive high-frequency noise $\mathcal{F}_n$ due to their random distribution property [14, 19], *i.e.*, $\mathcal{F} = \widetilde{\mathcal{F}} + \mathcal{F}_n$, where the clean field $\widetilde{\mathcal{F}}$ is estimated almost completely by inliers. Furthermore, due to the consistency of inliers, the motion vectors can be divided into finite clusters as $\{\boldsymbol{\mathcal{M}}^k|k = 1, \dots, K\}$ [9], where K is the total number of clusters. Correspondences in each cluster share the same motion pattern [14], thus each cluster can form a highly smooth sub-field, together they recover the clean field $\widetilde{\mathcal{F}}$. Hence, the original motion field can be decomposed as:

$$ \mathcal{F} = \widetilde{\mathcal{F}} + \mathcal{F}_n = \textstyle\sum_{k=1}^{K} \mathcal{F}^k + \mathcal{F}_n, \tag{2} $$

where the summation term can also be deemed as the "low-frequency" part of $\mathcal{F}$. $\mathcal{F}^k$ means the highly smooth sub-field formed by motion vectors $\{\boldsymbol{m}_i^k | i = 1, \dots, |\boldsymbol{\mathcal{M}}^k|\}$ that belong to the k-th cluster $\boldsymbol{\mathcal{M}}^k$, and $\{\boldsymbol{m}_i^k\}$ share the same motion pattern $\boldsymbol{b}^k$. In other words, to generate $\mathcal{F}^k$, the motion vector $\boldsymbol{m}_i$ that subordinates to the same motion pattern $\boldsymbol{b}^k$ should be grouped into the same cluster:

$$\begin{cases} \boldsymbol{m}_i \in \boldsymbol{\mathcal{M}}^k, & \text{if} \quad \text{sim}(\boldsymbol{b}^k, \boldsymbol{m}_i) \geq \tau, \\ \boldsymbol{m}_i \notin \boldsymbol{\mathcal{M}}^k, & \text{otherwise.} \end{cases} \tag{3}$$

The function $\text{sim}(\boldsymbol{b}^k, \boldsymbol{m}_i)$ attempts to measure the similarity between motion vector $\boldsymbol{m}_i$ and motion pattern $\boldsymbol{b}^k$, and τ is a threshold. Eq. (3) illustrates that, if a motion vector $\boldsymbol{m}_i$ is more similar to a motion pattern $\boldsymbol{b}^k$, it will more likely belong to the cluster $\boldsymbol{\mathcal{M}}^k$ represented by that motion pattern. After such clustering, the highly smooth sub-field $\mathcal{F}^k$ can be obtained with cluster $\boldsymbol{\mathcal{M}}^k$:

$$\mathcal{F}^k = \phi(\boldsymbol{\mathcal{M}}^k) = \phi(\{\boldsymbol{m}_i^k | i = 1, \dots, |\boldsymbol{\mathcal{M}}^k|\}). \tag{4}$$

The interpolating function $\phi(\cdot)$ approximates the dense motion field with sparse motion vectors [2, 19]. With Eqs. (3) and (4), all smooth sub-fields can be expressed theoretically.

However, the hard assignment of motion vectors in Eq. (3) is not differentiable, leading to the gradient disappearance in the deep network. Hence, we treat the similarity $\text{sim}(\boldsymbol{b}^k, \boldsymbol{m}_i) \in [0, 1]$ as the weight for each motion vector to achieve soft assignment, thus rewriting Eq. (4) as:

$$\mathcal{F}^k = \phi(\{\text{sim}(\boldsymbol{b}^k, \boldsymbol{m}_i) \cdot \boldsymbol{m}_i | i = 1, \dots, N\}). \tag{5}$$

Therefore, the sub-field $\mathcal{F}^k$ can be calculated in a differentiable way. Furthermore, due to the highly smooth $\mathcal{F}^k$ is derived from the motion vectors of merely one motion pattern, we can assume that it depends on the motion vector $\boldsymbol{z}^k$ only, *i.e.*, $\mathcal{F}^k = \phi(\{\boldsymbol{z}^k\})$. Comparing it with Eq. (5):

$$\boldsymbol{z}^k = \textstyle\sum_{i=1}^{N} \text{sim}(\boldsymbol{b}^k, \boldsymbol{m}_i) \cdot \boldsymbol{m}_i, \;\; \text{sim}(\boldsymbol{b}^k, \boldsymbol{m}_i) \in [0, 1]. \tag{6}$$

From Eq. (6), the motion vector $\boldsymbol{z}^k$ which supports the sub-field $\mathcal{F}^k$ actually describes the projection of all motion vectors (*i.e.* the projection of motion field $\mathcal{F}$) on a particular motion pattern $\boldsymbol{b}^k$, thus $\boldsymbol{b}^k$ can be regarded as the basis vector and $\boldsymbol{z}^k$ is the coefficient of $\mathcal{F}$ on $\boldsymbol{b}^k$. In this way, original motion field $\mathcal{F}$ in Eq. (2) is decomposed into finite highly smooth sub-fields $\{\mathcal{F}^k\}$ while $\mathcal{F}_n$ caused by outliers is ignored naturally, and $\{\boldsymbol{z}^k\}$ are both the support motion vectors and the representations of $\{\mathcal{F}^k\}$ on the basis $\{\boldsymbol{b}^k\}$.

To implement Eq. (6) in a learnable pipeline, we convert $\boldsymbol{m}_i$ into high dimensional space with Eq. (1), and set $\{\boldsymbol{b}^k\}$ as learnable tokens with $\boldsymbol{b}^k \in \mathbb{R}^C$. Moreover, we add some learnable weights for every element in Eq. (6) and choose Softmax operation as the similarity function $\text{sim}(\cdot, \cdot)$, then Eq. (6) exhibits a strong resemblance to the attention mechanism [35], and we rewrite it with matrix form:

$$\boldsymbol{Z} = \mathcal{A}(\boldsymbol{B}, \boldsymbol{F}) = \text{Softmax}\left(\frac{(\boldsymbol{W}_Q\boldsymbol{B})(\boldsymbol{W}_K\boldsymbol{F})^T}{\sqrt{C}}\right)\boldsymbol{W}_V\boldsymbol{F}, \tag{7}$$

where $\boldsymbol{W}_Q$, $\boldsymbol{W}_K$, $\boldsymbol{W}_V$ are learnable weights, and $\boldsymbol{B} \in \mathbb{R}^{K\times C}$, $\boldsymbol{F} \in \mathbb{R}^{N\times C}$, $\boldsymbol{Z} \in \mathbb{R}^{K\times C}$ are the matrix forms of $\{\boldsymbol{b}^k\}, \{\boldsymbol{f}_i\}, \{\boldsymbol{z}^k\}$ with complementary columns of all-one vectors as bases. $\mathcal{A}(\boldsymbol{B}, \boldsymbol{F})$ calculates the similarity scores between motion features and learnable basis, aggregating information from the motion field to pre-defined motion patterns. Practically, Eq. (7) is implemented by a stronger graph attention network (GAT) [36]. It realizes information aggregation among unordered data with standard attention including feed-forward network and shortcut connection:

$$\boldsymbol{Z} = \mathcal{G}(\boldsymbol{B}, \boldsymbol{F}) = \boldsymbol{B} + \text{FFN}(\boldsymbol{B} \| \mathcal{A}(\boldsymbol{B}, \boldsymbol{F})), \tag{8}$$

where $\mathcal{G}(\cdot, \cdot)$ indicates the GAT, $\|$ denotes concatenating by channels, and $\text{FFN}(\cdot)$ means feed-forward network (FFN) that compacts the result of concatenation to the same channels as $\boldsymbol{B}$. With Eq. (8), original motion vectors are clustered and the motion field is decomposed with the finite basis. Furthermore, referring to existent learning-based methods, we adopt the consistently used multi-layer progressive structure [17, 42, 43] to gradually filter out the outliers and improve the network performance. Hence, the decomposition of the motion field in layer ℓ is represented as:

$${}^{\ell}\boldsymbol{Z} = \mathcal{G}(\boldsymbol{B}, {}^{\ell}\boldsymbol{F}), \quad \ell = 0, \dots, L-1, \tag{9}$$

where L denotes the number of layers, $\boldsymbol{B} = \{\boldsymbol{b}^k\}$, ${}^{\ell}\boldsymbol{F} = \{{}^{\ell}\boldsymbol{f}_i\}$, ${}^{\ell}\boldsymbol{Z} = \{{}^{\ell}\boldsymbol{z}^k\}$ are the learnable basis, the motion vector features, and the sub-field representations in layer ℓ. Note that the basis $\{\boldsymbol{b}^k\}$ does not change from different layers to ensure the stable decomposition of ${}^{\ell}\mathcal{F}$, and the structure details of the GAT can be seen in the *S.M.*

Global Information Enhancement. Global information has been proven to play an important role in learning-based outlier rejection methods [41, 44]. Although the global information is extracted implicitly in Eq. (9), we still expect to further emphasize it. Thus, we update the representations of sub-fields by global self-attention with GAT in layer ℓ:

$${}^{\ell}\widetilde{\boldsymbol{Z}} = \mathcal{G}({}^{\ell}\boldsymbol{Z}, {}^{\ell}\boldsymbol{Z}), \quad \ell = 0, \dots, L-1. \tag{10}$$

Then global context is embedded into the representations of sub-fields as ${}^{\ell}\widetilde{\boldsymbol{Z}} = \{{}^{\ell}\widetilde{\boldsymbol{z}}^k\}$.

Masked Decomposition Strategy. As mentioned above, even though the negative influence of outliers seems to be eliminated during decomposing and clustering (*i.e.*, Eqs. (2)-(4)), the soft assignment scheme used in Eq. (5) is inevitably disturbed by outliers. So we try to remove the

known distinct outliers before the decomposition to mitigate the adverse influence. Specifically, once we obtain the inlier probability of all correspondences as ${}^{\ell}\boldsymbol{p} = \{{}^{\ell}p_i \,|i = 1, \ldots, N\}$ where ${}^{\ell}\boldsymbol{p} \in [0,1]^N$ is a probability vector predicted by the inlier predictor of layer $\ell - 1$ (the inlier predictor is described in Section 2.4), we expand the vector ${}^{\ell}\boldsymbol{p}$ to a matrix ${}^{\ell}\boldsymbol{P} \in [0,1]^{N\times C}$ by repeating along the column so that the masked decomposition in Eq. (9) of layer ℓ is:

$$ {}^{\ell}\boldsymbol{Z} = \mathcal{G}(\boldsymbol{B}, {}^{\ell}\boldsymbol{P} \odot {}^{\ell}\boldsymbol{F}), \quad \ell = 0, \ldots, L-1, \tag{11} $$

where $\odot$ is Hadamard product. The probability ${}^{\ell}p_i$ of outlier is 0, so that false matches contribute nothing to subfields and their representations. Note that for the first layer of DeMatch, there is no upper layer that provides the probability of matches, so we assume ${}^{0}\boldsymbol{p} = \mathbf{1}$.

By now, with Eqs. (7)-(11), the original motion field ${}^{\ell}\mathcal{F}$ has been decomposed into several highly smooth sub-fields, and the representations of sub-fields embedded by global context on the pre-defined basis are $\{{}^{\ell}\widetilde{\boldsymbol{z}}^k\}$. Subsequently, a cleaner motion field should be recovered with these "low-frequency" sub-fields, so that motion vectors of outliers are corrected and the piecewise smoothness can be guaranteed.

2.3. Recovery of A Cleaner Motion Field

Acting as the respectively unique support motion vectors of sub-fields, motion representations $\{{}^{\ell}\widetilde{\boldsymbol{z}}^k\}$ can be obtained from motion vector features $\{{}^{\ell}\boldsymbol{f}_i\}$ with Eqs. (6)-(8). Thus new motion vector features ${}^{\ell+1}\boldsymbol{F} = \{{}^{\ell+1}\boldsymbol{f}_i\}$ can be also derived from the support motion vectors in turn:

$$ {}^{\ell+1}\boldsymbol{F} = \mathcal{G}({}^{\ell}\boldsymbol{F}, {}^{\ell}\widetilde{\boldsymbol{Z}}), \quad \ell = 0, \ldots, L-1. \tag{12} $$

Note that ${}^{0}\boldsymbol{F}$ is obtained from Eq. (1). The new high dimensional motion vectors ${}^{\ell+1}\boldsymbol{F}$ are the input of the next layer and can be easily utilized to describe a recovered motion field. So with Eq. (12), a smooth field ${}^{\ell+1}\mathcal{F}$ is recovered implicitly with the representations of "low-frequency" highly smooth sub-fields, and the new motion vector ${}^{\ell+1}\boldsymbol{f}_i$ derived from ${}^{\ell+1}\mathcal{F}$ is corrected relative to ${}^{\ell}\boldsymbol{f}_i$.

2.4. Prediction of Inliers

The corrected motion vector ${}^{\ell+1}\boldsymbol{f}_i$ can substitute ${}^{\ell}\boldsymbol{f}_i$ after the decomposition and recovery. By the rule that motion vectors of inliers should not change a lot after correcting but outliers change prominently, we can compare the difference between them to classify inliers and outliers with a threshold. Refer to [42, 43], an inlier predictor is also used in each layer of DeMatch. The input of predictor is $\{{}^{\ell+1}\boldsymbol{f} - {}^{\ell}\boldsymbol{f}\}$ while output is predicted logits ${}^{\ell}\hat{\boldsymbol{\omega}} = \{{}^{\ell}\hat{\omega}_i\}$ and inlier probability ${}^{\ell+1}\boldsymbol{p} = \max(0, \tanh {}^{\ell}\hat{\boldsymbol{\omega}})$. Details of the inlier predictor are shown in the *S.M.*

2.5. Loss Functions

We choose a widely used loss function [17, 41–43] as:

$$ \mathcal{L} = \textstyle\sum_{\ell=0}^{L-1} \mathcal{L}_{cls}(\boldsymbol{\omega}, {}^{\ell}\hat{\boldsymbol{\omega}}) + \lambda \mathcal{L}_{reg}(\boldsymbol{E}, {}^{\ell}\widehat{\boldsymbol{E}}), \tag{13} $$

where $\mathcal{L}$ includes classification loss $\mathcal{L}_{cls}$ and regression loss $\mathcal{L}_{reg}$, λ is a hyper-parameter to balance them. $\mathcal{L}_{cls}$ is a binary cross entropy loss, $\boldsymbol{\omega}$ is the weakly supervised labels judged by a threshold (*e.g.* 10^{-4}) with Sampson distance [12]. $\mathcal{L}_{reg}$ is also obtained with the Sampson distance:

$$ \mathcal{L}_{reg}(\boldsymbol{E}, \widehat{\boldsymbol{E}}) = \sum_{i=1}^{N} \frac{(\boldsymbol{y}_i^T \widehat{\boldsymbol{E}} \boldsymbol{x}_i)^2}{\|\boldsymbol{E}\boldsymbol{x}_i\|_{[1]}^2 + \|\boldsymbol{E}\boldsymbol{x}_i\|_{[2]}^2 + \|\boldsymbol{E}^T\boldsymbol{y}_i\|_{[1]}^2 + \|\boldsymbol{E}^T\boldsymbol{y}_i\|_{[2]}^2}, \tag{14} $$

where $\widehat{\boldsymbol{E}}$ is the essential matrix calculated by the weighted eight-point algorithm [41] with inlier probability $\boldsymbol{p}$, $\boldsymbol{E}$ is ground truth, and $\|\boldsymbol{\nu}\|_{[i]}$ is the i-th element of vector $\boldsymbol{\nu}$.

2.6. Implementation Details

For implementation, we normalize the coordinates of keypoints to $[-1, 1]$ with image size or camera intrinsic. We stack the Coherence Constraint Layer 5 times (*i.e.*, $L = 5$), and conduct Eq. (9) 2 times and Eq. (10) 4 times in a layer to enhance the representations of sub-fields and the embedding of global information. We use Adam [13] for training with a learning rate of 10^{-4} during the first $80k$ iterations then decaying with factor 0.999996 every step. We stop the training at $500k$ iterations for outdoor scenes and $700k$ iterations for indoor scenes. The batch size is set as 32. λ is 0 during the first $20k$ iterations and then 0.5 [42]. All training and testing are performed with a single RTX3090 GPU.

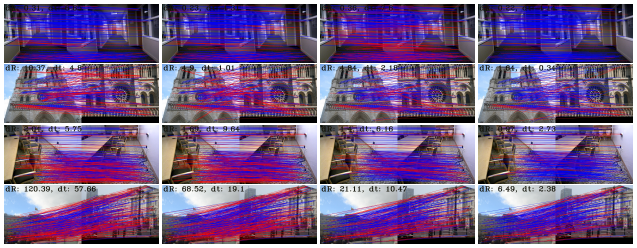
3. Experiments

3.1. Relative Pose Estimation

Many vision applications include the estimation of positional relationships (rotation and translation) between the cameras that capture image pairs in the same scene. The accuracy of estimation heavily depends on the quality of predicted inliers. We evaluate DeMatch on this task with both outdoor and indoor scenes. Refer to the settings in [42], we use outdoor YFCC100M [32] and indoor SUN3D [37] datasets, and detect up to $2k$ keypoints with SIFT [18], building the putative set with the nearest neighbor (NN) method. To evaluate estimation accuracy, we report the area under the cumulative error curve (**AUC**) of the maxima pose error of rotation and translation at multiple thresholds $(5°, 10°, 20°)$ [41, 42]. We choose the NN method as a baseline which retains all putative correspondences. Then we compare DeMatch with several classical outlier rejection methods [3, 9, 19, 20], together with a few learning-

Table 1. Relative pose estimation results with the weighted eight-point algorithm / RANSAC. **AUC** at 5°, 10°, 20° is reported respectively.

Method	YFCC100M [32]			SUN3D [37]		
	@5°	@10°	@20°	@5°	@10°	@20°
NN	– / 3.47	– / 9.10	– / 18.60	– / 1.04	– / 3.43	– / 8.75
GMS [3]	– / 13.29	– / 24.38	– / 37.83	– / 4.12	– / 10.53	– / 20.82
LPM [20]	– / 15.99	– / 28.25	– / 41.76	– / 4.80	– / 12.28	– / 23.77
CRC [9]	– / 16.51	– / 28.01	– / 41.38	– / 4.07	– / 10.44	– / 20.87
VFC [19]	– / 17.43	– / 29.98	– / 43.00	– / 5.26	– / 13.05	– / 24.84
PointCN [41]	10.16 / 26.73	24.43 / 44.01	43.31 / 60.49	3.05 / 6.09	10.00 / 15.43	24.06 / 29.74
OANet [42]	15.92 / 27.26	35.93 / 45.93	57.11 / 63.17	5.93 / 6.78	16.91 / 17.10	34.32 / 32.41
CLNet [44]	24.34 / 31.45	44.69 / 51.06	63.61 / 68.40	1.55 / 6.67	5.11 / 16.81	13.61 / 31.45
MS^{2}DGNet [6]	20.61 / 31.55	42.90 / 50.94	64.26 / 68.34	5.88 / 7.13	16.83 / 17.80	34.28 / 33.47
NCMNet [16]	26.89 / 32.30	46.19 / 52.29	64.21 / **69.65**	6.31 / 7.10	16.84 / 18.56	33.11 / **35.58**
LMCNet [17]	22.35 / 30.48	43.57 / 49.84	63.34 / 66.94	7.08 / 6.84	19.09 / 17.62	37.15 / 33.43
ConvMatch [43]	26.83 / 31.69	49.14 / 51.41	67.91 / 68.45	8.76 / 7.32	22.23 / 18.45	40.49 / 34.41
DeMatch (Ours)	**30.89 / 32.98**	**52.67 / 52.37**	**70.33** / 69.01	**9.31 / 7.44**	**23.10 / 18.66**	**41.55** / 34.78



OANet LMCNet ConvMatch DeMatch (Ours)

Figure 3. Qualitative illustration of outlier rejection. Mark false matches with red while correct matches with blue. The relative pose estimation results (error of rotation and translation) are provided in the top left corner. Zoom in for better visualization.

based methods [6, 16, 41, 42, 44] that classify correspondences directly, and [17, 43] that consider coherence and smoothness property. During the evaluation, we try to use different robust model estimators including the weighted eight-point algorithm [41] and RANSAC [10] to calculate the positional relationship from predicted inliers. Note that the weighted eight-point algorithm requires the inlier probability of each correspondence thereby being unsuitable for classical methods. All the results are shown in Table 1, DeMatch outperforms all other methods. We further illustrate the qualitative results of outlier rejection and relative pose estimation in Figure 3. The excellent performance reveals that the motion coherence-based methods determine inliers more inherently than the directly classifying methods, and that decomposing the motion field on a finite basis is a better approach to exploit the prior knowledge of coherence and smoothness, especially in the case of large scene disparities. More visualization results are shown in the *S.M.*

3.2. Visual Localization

Advances in correspondence learning also benefit practical issues such as visual localization [25, 28], which aims to estimate a 6-DOF position of a query image with respect to a 3D model. This task is strongly challenged by complicated conditions such as viewpoint and illumination changes, thus accurate outlier rejection is required. Following [5], we integrate our method into the official HLoc [25] pipeline for visual localization on the Aachen Day-Night benchmark [27, 28]. Particularly, we extract up to $4k$ keypoints per image with SIFT [18], match them with the mutual nearest neighbor (MNN) method and remove outliers with correspondence learning methods, triangulate an SfM model from day-time images with known pose, and register both day-time and night-time query images from the predicted inliers with COLMAP [29]. We select almost the same compared methods as relative pose estimation. All models are trained with SIFT features on YFCC100M [32] dataset. Table 2 reports the pose estimation accuracy of different methods on the visual localization task. DeMatch performs better, especially in the difficult night-time scenes.

Table 2. Visual localization results.

Method	Day	Night
	(0.25m,2°) / (0.5m, 5°) / (5.0m, 10°)	
PointCN [41]	83.1 / 92.2 / 96.2	69.4 / 79.6 / 89.8
OANet [42]	83.1 / 92.5 / 96.6	72.4 / 80.6 / 90.8
CLNet [44]	83.3 / 92.4 / 97.0	71.4 / 80.6 / 93.9
MS^{2}DGNet [6]	83.4 / 92.7 / **97.2**	72.4 / 82.7 / 92.9
LMCNet [17]	84.1 / **92.8** / 97.1	71.4 / 81.6 / 93.9
ConvMatch [43]	84.5 / 92.7 / 96.8	**73.5** / 83.7 / 91.8
DeMatch (Ours)	**85.2 / 92.8** / 97.1	**73.5 / 84.7 / 94.9**

3.3. Analysis

We further analyze DeMatch in this section. First, we calculate the computational usage to show that the decomposition in DeMatch is more efficient than explicit regularization, and display the natural piecewise smoothness property through appropriate visualization. Both of them testify that DeMatch does solve the problems in existing methods.

Table 3. Computational usage from four different aspects.

Method	Time (ms)	Flops (G)	Param (M)	Mem (MB)
PointCN [41]	8.52	1.181	0.595	7.54
OANet [42]	13.91	1.841	2.473	20.64
CLNet [44]	20.55	1.921	0.953	87.78
MS^{2}DGNet [6]	16.63	5.044	2.613	95.88
LMCNet [17]	227.15	–	0.925	25.08
ConvMatch [43]	24.65	3.783	7.494	51.49
DeMatch (Ours)	26.36	2.346	5.853	41.97

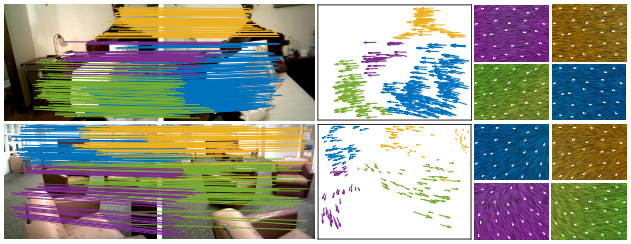


Figure 4. Illustration of piecewise smoothness.

Table 4. Apply DeMatch after different matchers. The metric is **AUC** with different geometric model estimators.

Matcher	Filter	Estimator	@5°	@10°	@20°
SP+SG [7, 26]	–	DLT	18.87	32.92	48.87
	DeMatch		30.71	51.38	70.11
	–	RANSAC [10]	38.06	58.38	74.67
	DeMatch		39.90	60.65	76.51
SP+LG [7, 15]	–	DLT	26.65	42.62	58.89
	DeMatch		31.60	52.38	71.04
	–	RANSAC	39.42	59.69	75.89
	DeMatch		40.53	60.72	76.68
LoFTR [31]	–	DLT	5.24	13.24	26.40
	DeMatch		20.97	40.38	60.69
	–	RANSAC	39.80	60.03	76.07
	DeMatch		40.84	61.11	76.71
DKM [8]	–	DLT	29.02	44.90	59.75
	DeMatch		32.14	49.00	66.56
	–	RANSAC	44.10	63.75	78.42
	DeMatch		45.14	64.99	79.45

Then we prove its compatibility with matchers, evaluate the generalization ability on different datasets and descriptors, determine the parameters of the network structure, test the effect of different initializations of the learnable basis, and perform ablation studies to reveal the effectiveness of several components and strategies in DeMatch.

Computational Usage. We calculate the computational usage, including average running time (Time), floating point operations (Flops), model parameter size (Param), and peak memory usage (Mem). Note that the average running time is the running time of the algorithm per image on YFCC100M [32]. Table 3 shows the results. In particular, LMCNet costs a lot due to its graph construction and matrix decomposition for solving the explicit regularization term [17], while DeMatch runs much faster with implicit regularization during the decomposition of the motion field. Compared to other methods, DeMatch is able to perform better with comparable computational usage.

Piecewise Smoothness Property. To demonstrate that the piecewise smoothness property of DeMatch is generated naturally with the decomposition of the motion field, we illustrate the correspondence and motion clusters during the assignment process of Eq. (3) with the similarity scores in Eq. (7). As illustrated in Figure 4, the correspondences (the left column) and the motion vectors (the middle column) are grouped in different colors, each cluster is highly consistent and subordinates to a particular motion pattern. The formed sub-fields are highly smooth (the right column) although different sub-fields follow different motion flows, hence the motion field recovered by these sub-fields that respectively belong to unique motion flows is obviously piecewise smooth, maintaining discontinuities at the edge of different sub-fields correctly. Note that we range all the clusters by the number of correspondences, and draw the top-4 clusters for better visualization. More results are shown in the *S.M.*

Compatibility with Matchers. We evaluate the performance of relative pose estimation task with or without DeMatch on the YFCC100M [32] dataset using different widely used matchers, including SuperPoint [7] paired with SuperGlue [26] (noted as SP+SG), SuperPoint paired with LightGlue [15] (noted as SP+LG), LoFTR [31], and DKM [8], while the pose estimator is Direct Linear Transform (DLT) or RANSAC [10]. For SP+SG and SP+LG, we follow the settings of SuperGlue, detecting up to $1k$ keypoints. For LoFTR and DKM, the evaluation pipeline is almost as the same as [33]. Note that with the suggestions of the same experiment in [17], we do not adopt the filtering strategy in each method itself and retain all putative correspondences as input. Results in Table 4 reveal that, as a general method for outlier filtering, DeMatch can further bring improvement over the state-of-the-art matchers and can serve as a complementary module for practical usage.

Generalization Ability. Generalization means applying the same model to different descriptors and scenes. It is common that the model learned from SIFT [18] performs poorly on other descriptors, or the model that performs well in outdoor scenes almost fails in indoor scenes. In general, motion coherence-based methods display better generalization ability than the directly classifying methods for solving outlier rejection more naturally [43], including DeMatch in our paper. To demonstrate it, we repeat the relative pose estimation on YFCC100M [32] with RootSIFT [1], LIFT [40] and SuperPoint [7], and on SUN3D [37] with SIFT, RootSIFT, LIFT and SuperPoint, employing the same model trained on YFCC100M with SIFT. We detect up to $2k$ keypoints, and use an NN method to generate the putative set. Results are

Table 5. Generalization ability test. **AUC**@5° with the weighted eight-point algorithm is reported.

Method	YFCC100M [32]			SUN3D [37]			
	RootSIFT [1]	LIFT [40]	SuperPoint [7]	SIFT [18]	RootSIFT	LIFT	SuperPoint
PointCN [41]	10.45	5.05	8.33	0.35	0.42	0.49	1.08
OANet [42]	16.09	11.38	13.49	0.89	0.98	0.86	1.00
CLNet [44]	24.62	15.47	22.39	0.75	0.80	0.66	1.01
MS^{2}DGNet [6]	21.16	14.97	16.44	1.60	1.74	1.21	1.50
LMCNet [17]	22.90	15.71	18.36	1.64	1.69	1.17	1.59
ConvMatch [43]	27.53	17.75	23.74	2.19	2.37	2.03	**2.17**
DeMatch (Ours)	**31.34**	**21.92**	**27.91**	**2.21**	**2.48**	**2.09**	2.12

Table 6. Parameter settings. Report **AUC**@10° with the weighted eight-point algorithm. One parameter is fixed while another varies.

Metric	L=3	$\boldsymbol{L}$**=5**	L=7	K=32	$\boldsymbol{K}$**=48**	K=64
AUC@10°	45.38	52.67	53.05	49.85	52.67	51.79
Flops (G)	1.425	2.346	3.268	2.270	2.346	2.423
Param (M)	3.520	5.853	8.185	5.853	5.853	5.853

Table 7. Different initialization methods for $\boldsymbol{B}$. Results of **AUC** on relative pose estimation with RANSAC are reported.

Initialization	@5°	@10°	@20°
All-zeros	30.83	49.96	66.86
All-ones	30.94	50.16	67.14
Normal (default)	32.98	52.37	69.01
Uniform	33.73	53.16	69.70
Kaiming-normal	33.15	52.58	69.36
Kaiming-uniform	33.48	53.06	69.68

shown in Table 5. DeMatch displays the consistently excellent generalization of motion coherence-based methods.

Parameter Settings. It is important to determine the number of Coherence Constraint Layers L and of motion patterns K. Theoretically, a larger L is conducive to filtering out more outliers while leading to higher computational usage, but a suitable value of K should be chosen because too few base vectors may result in an over-smooth motion field and too many base vectors are just redundant where outliers may be involved. We finally choose $L = 5, K = 48$ to achieve performance and consumption balance. The results of outdoor relative pose estimation in Table 6 with different L and K can support our choice. We further analyze why it is a good choice in the *S.M.* We also examine the effect of the number of times Eqs. (9) and (10) are reused in each layer on the performance of DeMatch in the *S.M.*

Initialization of Basis. We try different initializing distributions for the learnable basis $\boldsymbol{B}$, including all-zeros, all-ones, uniform, normal (the default method), kaiming-uniform, and kaiming-normal. We repeat the relative pose estimation and results are shown in Table 7. Expect the all-zeros and all-ones, the results are not that different. Thus, any reasonable initialization of $\boldsymbol{B}$ can give a good result.

Ablation Studies. We conduct ablation studies by repeating relative pose estimation. Results are shown in Table 8. (**i**) is the full Dematch with standard attention in Eq. (8) and the decomposition process, including its masked decomposition strategy and global information enhancement module. From (**ii**) to (**v**), we eliminate the components of the decomposition process and even itself, the performance drops progressively. And (**vi**) chooses vanilla attention in Eq. (7), performing worse than (**i**). Table 8 reveals that DeMatch benefits from all the ingredients mentioned above.

Table 8. Ablation studies. "Att." means the attention type, including standard type "S." and vanilla type "V.". "De." means the decomposition process, "Mask" means the masked decomposition strategy, "Global" means the global information enhancement module. **AUC** with the weighted eight-point algorithm is reported.

Num.	Att.	De.	Mask	Global	@5°	@10°	@20°
(i)	S.	✓	✓	✓	**30.89**	**52.67**	**70.33**
(ii)	S.	✓		✓	29.60	51.10	69.19
(iii)		✓	✓		25.29	46.55	65.48
(iv)		✓			22.39	43.64	63.47
(v)					18.23	38.48	59.15
(vi)	V.	✓	✓	✓	28.97	50.93	69.30

4. Conclusion

We design a novel network called DeMatch for two-view correspondence learning. Inspired by Fourier expansion, DeMatch tries to constrain the coherence of the motion field by retaining the main "low-frequency" and smooth part, decomposing the contaminated motion field in deep space. By choosing a finite basis that describes a few motion patterns, motion vectors are clustered while outliers are removed, and the potential field is accordingly decomposed into several highly smooth sub-fields. The finite decomposition can be regarded as an implicit regularization term, achieving lower computational usage, and the recovery of the cleaner field with these sub-fields generates piecewise smoothness naturally. Extensive experiments demonstrate the superiority of our method and the promising properties mentioned above.

Acknowledgments

This work was supported by NSFC (62276192).

References

[1] Relja Arandjelović and Andrew Zisserman. Three things everyone should know to improve object retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2911–2918, 2012. 7, 8

[2] Luca Baldassarre, Lorenzo Rosasco, Annalisa Barla, and Alessandro Verri. Multi-output learning via spectral filtering. *Machine Learning*, 87:259–301, 2012. 4

[3] JiaWang Bian, Wen-Yan Lin, Yasuyuki Matsushita, Sai-Kit Yeung, Tan-Dat Nguyen, and Ming-Ming Cheng. Gms: Grid-based motion statistics for fast, ultra-robust feature correspondence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4181–4190, 2017. 2, 5, 6

[4] Matthew Brown and David G Lowe. Automatic panoramic image stitching using invariant features. *International Journal of Computer Vision*, 74:59–73, 2007. 1

[5] Hongkai Chen, Zixin Luo, Jiahui Zhang, Lei Zhou, Xuyang Bai, Zeyu Hu, Chiew-Lan Tai, and Long Quan. Learning to match features with seeded graph matching network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6301–6310, 2021. 6

[6] Luanyuan Dai, Yizhang Liu, Jiayi Ma, Lifang Wei, Taotao Lai, Changcai Yang, and Riqing Chen. Ms2dg-net: Progressive correspondence learning via multiple sparse semantics dynamic graph. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8973–8982, 2022. 2, 6, 7, 8

[7] 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, 7, 8

[8] Johan Edstedt, Ioannis Athanasiadis, Mårten Wadenbäck, and Michael Felsberg. Dkm: Dense kernelized feature matching for geometry estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 17765–17775, 2023. 7

[9] Aoxiang Fan, Xingyu Jiang, Yong Ma, Xiaoguang Mei, and Jiayi Ma. Smoothness-driven consensus based on compact representation for robust feature matching. *IEEE Transactions on Neural Networks and Learning Systems*, 2023. 1, 2, 3, 5, 6

[10] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 6, 7

[11] Paul R. Halmos. *Finite dimensional vector spaces*. Princeton University Press, 1947. 2

[12] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2003. 5

[13] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representation*, 2015. 5

[14] Wen-Yan Lin, Fan Wang, Ming-Ming Cheng, Sai-Kit Yeung, Philip HS Torr, Minh N Do, and Jiangbo Lu. Code: Coherence based decision boundaries for feature correspondence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(1):34–47, 2018. 2, 3

[15] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. Lightglue: Local feature matching at light speed. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 17627–17638, 2023. 7

[16] Xin Liu and Jufeng Yang. Progressive neighbor consistency mining for correspondence pruning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9527–9537, 2023. 2, 6

[17] Yuan Liu, Lingjie Liu, Cheng Lin, Zhen Dong, and Wenping Wang. Learnable motion coherence for correspondence pruning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3237–3246, 2021. 2, 3, 4, 5, 6, 7, 8

[18] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 1, 5, 6, 7, 8

[19] Jiayi Ma, Ji Zhao, Jinwen Tian, Alan L Yuille, and Zhuowen Tu. Robust point matching via vector field consensus. *IEEE Transactions on Image Processing*, 23(4):1706–1721, 2014. 1, 3, 4, 5, 6

[20] Jiayi Ma, Ji Zhao, Junjun Jiang, Huabing Zhou, and Xiaojie Guo. Locality preserving matching. *International Journal of Computer Vision*, 127(5):512–531, 2019. 2, 5, 6

[21] Jiayi Ma, Xingyu Jiang, Aoxiang Fan, Junjun Jiang, and Junchi Yan. Image matching from handcrafted to deep features: A survey. *International Journal of Computer Vision*, 129(1):23–79, 2021. 1

[22] Anastasiia Mishchuk, Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. In *Proceedings of the Advances in Neural Information Processing Systems*, 2017. 1

[23] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015. 1

[24] Yuki Ono, Eduard Trulls, Pascal Fua, and Kwang Moo Yi. Lf-net: Learning local features from images. In *Proceedings of the Advances in Neural Information Processing Systems*, 2018. 1

[25] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12716–12725, 2019. 6

[26] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4938–4947, 2020. 7

[27] Torsten Sattler, Tobias Weyand, Bastian Leibe, and Leif Kobbelt. Image retrieval for image-based localization revisited. In *Proceedings of the British Machine Vision Conference*, pages 1–12, 2012. 6

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

[29] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4104–4113, 2016. 6

[30] Noah Snavely, Steven M Seitz, and Richard Szeliski. Modeling the world from internet photo collections. *International Journal of Computer Vision*, 80(2):189–210, 2008. 1

[31] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8922–8931, 2021. 7

[32] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016. 5, 6, 7, 8

[33] Prune Truong, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning accurate dense correspondences and when to trust them. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5714–5724, 2021. 7

[34] Michał Tyszkiewicz, Pascal Fua, and Eduard Trulls. Disk: Learning local features with policy gradient. In *Proceedings of the Advances in Neural Information Processing Systems*, 2020. 1

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the Advances in Neural Information Processing Systems*, 2017. 3, 4

[36] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *Proceedings of the International Conference on Learning Representation*, 2018. 4

[37] Jianxiong Xiao, Andrew Owens, and Antonio Torralba. Sun3d: A database of big spaces reconstructed using sfm and object labels. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1625–1632, 2013. 5, 6, 7, 8

[38] Weilong Yan, Robby T Tan, Bing Zeng, and Shuaicheng Liu. Deep homography mixture for single image rolling shutter correction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9868–9877, 2023. 2

[39] Nianjin Ye, Chuan Wang, Haoqiang Fan, and Shuaicheng Liu. Motion basis learning for unsupervised deep homography estimation with subspace projection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 13117–13125, 2021. 2

[40] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. Lift: Learned invariant feature transform. In *Proceedings of the European Conference on Computer Vision*, pages 467–483, 2016. 7, 8

[41] Kwang Moo Yi, Eduard Trulls, Yuki Ono, Vincent Lepetit, Mathieu Salzmann, and Pascal Fua. Learning to find good correspondences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2666–2674, 2018. 2, 4, 5, 6, 7, 8

[42] Jiahui Zhang, Dawei Sun, Zixin Luo, Anbang Yao, Lei Zhou, Tianwei Shen, Yurong Chen, Long Quan, and Hongen Liao. Learning two-view correspondences and geometry using order-aware network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5845–5854, 2019. 2, 4, 5, 6, 7, 8

[43] Shihua Zhang and Jiayi Ma. Convmatch: Rethinking network design for two-view correspondence learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3472–3479, 2023. 2, 3, 4, 5, 6, 7, 8

[44] Chen Zhao, Yixiao Ge, Feng Zhu, Rui Zhao, Hongsheng Li, and Mathieu Salzmann. Progressive correspondence pruning by consensus learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6464–6473, 2021. 2, 4, 6, 7, 8