DeMo: Deep Motion Field Consensus with Learnable Kernels for Two-view Correspondence Learning

Yifan Lu[1*], Jiajun Le[1*], Zizhuo Li[1], Yixuan Yuan[2], Jiayi Ma[1†]

[1]Electronic Information School, Wuhan University, Wuhan 430072, China
[2]Department of Electronic Engineering, Chinese University of Hong Kong, Sha Tin, Hong Kong SAR, China
lyf048@whu.edu.cn, jiajunle01@gmail.com, zizhuo_li@whu.edu.cn, yxyuan@ee.cuhk.edu.hk, jyma2010@gmail.com

Abstract

As a long-range prior, motion consensus essentially forces the overall spatial transformation between a pair of images to be smooth and consistent, which is naturally well-suited for two-view correspondence learning. However, such precious property remains under-explored by most existing studies due to the modeling challenges posed by the sparsity and uneven distributions of putative correspondences. In this paper, we propose DeMo, a novel and cutting-edge network for outlier rejection, which possesses the capacity to fully capture global motion consensus clues by way of consensus interpolation over the entire high-dimensional motion field generated by putative correspondences. Specifically, through incorporating regularization techniques into a Reproducing Kernel Hilbert Space (RKHS), a concise interpolation formula can be derived for the high-dimensional motion field, which inherently allows a closed-form solution. Subsequently, learnable deep kernels are collaboratively used to flexibly and efficiently capture the relationships between global inputs, thus maintaining the entire motion field consensus. In addition, to remedy the cubic computational overhead of explicit interpolation, a scene-adaptive sampling strategy is introduced, which implicitly selects the more scene-representative motions, reducing the computational complexity of motion consensus interpolation to be approximately linear while maintaining the accuracy. Moreover, to deal with underlying depth discontinuities caused by complicated scene variations, a local consensus complementation block is designed, which maintains local bilateral consensus across both feature and spatial channels. Without bells and whistles, DeMo achieves superior performance in various geometric tasks, including relative pose estimation, homography estimation, and visual localization.

Code — https://github.com/JiajunLe/DeMo

Introduction

Identifying reliable correspondences and estimating the relative pose between two images depicting the same scene captured from different perspectives, is a long-standing problem in computer vision, with applications to many high-level vision tasks, such as SLAM (Mur-Artal, Montiel, and Tardos 2015), visual localization (Philbin et al. 2010), and image fusion (Xu et al. 2022). The standard pipeline generally begins with detecting keypoints and generating local descriptors, followed by establishing putative correspondences based on the similarity between descriptors. Significant progress has been made in both traditional handcrafted descriptors and learning-based descriptors, such as SIFT (Lowe 2004) and SuperPoint (DeTone, Malisiewicz, and Rabinovich 2018). However, attributed to the ambiguity of visual descriptors caused by complex scenes such as viewpoint changes and illumination variations, putative correspondences inevitably contain considerable mismatches, which would induce significant deviations in geometry estimation. This paper focuses on rejecting outliers from putative correspondences.

Outlier rejection aimed at creating robust two-view correspondences has been explored long ago. As a long-range prior, motion consensus is a crucial cue for correspondence pruning, which means the overall spatial transformation between a pair of images is smooth and consistent. The classical method VFC (Ma et al. 2014) seeks motion consensus by estimating the vector field, while LPM (Ma et al. 2019) and GMS (Bian et al. 2017) focus on local consensus. However, traditional methods tend to fail when the putative set contains a significant number of outliers. Fortunately, the rapid development of deep learning has provided new technological breakthroughs for correspondence learning. Representatively, PointCN (Yi et al. 2018), for the first time, processes each correspondence individually and predicts inlier scores within an MLP framework, introducing context normalization to aggregate global information. But it overlooks the underlying local geometric relations. Thus, OANet (Zhang et al. 2019) designs a permutation-invariant pooling-like operation to capture local context. CLNet (Zhao et al. 2021) constructs annular convolution within a k-nearest neighbor graph to aggregate local context. Further, UMatch (Li, Zhang, and Ma 2023) proposes a multi-level hierarchy-aware framework to flexibly encode and decode high-level features for implicit local context. Even though these methods are effective, they neglect the property motion consensus which is inherently a highly important cue for rejecting spurious matches from candidate ones.

Essentially, the two-view correspondence learning task can be regarded as a motion field learning problem and recent advances have increasingly focused on this perspec-

*These authors contributed equally.
†Corresponding author.

tive, leading to notable performance improvement. Even so, the prevailing approaches have their focus solely confined to the narrow paradigm of local motion consensus. For instance, LMCNet (Liu et al. 2021) constrains motion consensus fitting to k-nearest neighbors, thereby discarding the global motion consensus, and suffers from extremely high computational consumption during graph construction. ConvMatch (Zhang and Ma 2024) employs CNNs to directly aggregate local context within an ordered vector field, whereas the CNN blocks cannot capture global motion consensus and lead to excessive smoothing. Recently, to address the discontinuities in the motion field, DeMatch (Zhang et al. 2024) decomposes the initial motion field into multiple smooth subfields, each with a different motion pattern. From a global perspective, the overall spatial transformation between an image pair should be consistent, emphasizing the importance of global motion consensus that is commensurate with local motion consensus. It is evident that existing approaches fail to ensure consensus across the entire motion field. At this point, one might question: *Is it possible to design a network that aggregates both local and* ***global*** *motion consensus, while considering their compatibility and ensuring the handling of discontinuities in the motion field?*

To positively answer the above question, we propose an innovative network for two-view correspondence learning, dubbed **De**ep **Mo**tion Field Consensus Network (DeMo), which has the capability of capturing global motion consensus in high-dimensional motion fields with learnable kernels. To be specific, we exploit regularization techniques in the RKHS to derive interpolation formulas and provide closed-form solutions for the high-dimensional motion fields. Within this framework, learnable deep kernels are introduced for deep motion consensus interpolation, therefore learning the relationships between global inputs flexibly and effectively. Considering the high computational complexity of the explicit interpolation manner (i.e., $\mathcal{O}(N^3)$), we propose a scene-adaptive sampling strategy that reduces the complexity from $\mathcal{O}(N^3)$ to nearly $\mathcal{O}(N)$ by implicitly selecting motions that are more scene-representative while retaining as much scene information as possible. In addition, the presence of discontinuities in motion field caused by large disparities, such as multiple objects located at different depths, is not negligible. To cope with this issue, we innovatively design a local consensus complementation (LCC) block, which complements consensus between spatial and feature channels, to identify changes in the motion field caused by large depth differences and avoid over-smoothing. With all the above ingredients, DeMo outperforms the state-of-the-arts on multiple highly challenging benchmarks. Our contributions can be summarized as follows:

- We innovatively propose DeMo for two-view correspondence learning, which exploits regularization in RKHS to fully capture motion field consensus in high-dimensional motion fields with learnable kernels.
- A scene-adaptive sampling strategy is introduced, which implicitly select the more scene-representative motions, and can reduce the complexity of explicit interpolation manner (i.e., $\mathcal{O}(N^3)$) to nearly $\mathcal{O}(N)$.
- To cope with the discontinuities in the motion field, we design an LCC block, which complements consensus between spatial and feature channels, thereby distinguishing discontinuities and avoiding over-smoothing.
- DeMo achieves state-of-the-art results across different benchmarks, showcasing robust generalization capabilities across diverse datasets and descriptors.

Method

Given two images $\mathbf{I}_A$ and $\mathbf{I}_B$ capturing the same scene from different viewpoints, we first extract keypoints and establish the putative correspondences $\{(\mathbf{x}_i, \mathbf{y}_i)|i = 1, \dots, N, \mathbf{x}_i \in \mathbb{R}^2, \mathbf{y}_i \in \mathbb{R}^2\}$ using the nearest neighbor (NN) matching algorithm, where $\mathbf{x}_i$ and $\mathbf{y}_i$ represent the coordinates of two corresponding keypoints that are normalized by camera intrinsics. The architecture is shown in Figure 1. The input is the putative motion vectors $\{\mathbf{m}_i = (\mathbf{x}_i, \mathbf{d}_i)|i = 1, \dots, N, \mathbf{m}_i \in \mathbb{R}^4\}$, where $\mathbf{d}_i = \mathbf{y}_i - \mathbf{x}_i$ denotes the displacement. Specifically, we first encode the motion vectors $\{\mathbf{m}_i\}_{i=1}^N$ and the coordinates of initial points $\{\mathbf{x}_i\}_{i=1}^N$ into a high-dimensional space, then input them into the first layer of the network. Next, we implement context aggregation sequentially on the feature and spatial channels in the local neighborhoods, with the latter serving as the consensus complement to the former. Additionally, we apply ContextNorm (Yi et al. 2018) to mitigate the feature space discrepancies between local and global modules caused by their separate processing. In the high-dimensional feature space, we utilize regularization techniques in RKHS (Micchelli and Pontil 2005; Aronszajn 1950; Baldassarre et al. 2012) with learnable kernels to implement interpolation across the entire motion field, thus fully capturing global motion consensus to rectify the motion field. Finally, the rectified motion vectors are employed to predict the inlier/outlier classification results. To leverage the powerful representation capabilities of deep neural networks, DeMo is stacked with L layers. Next, we present the novel network DeMo in detail.

Initialization

The motion vectors $\{\mathbf{m}_i\}_{i=1}^N$ will serve in the motion field interpolation and inlier prediction. However, effectively utilizing deep neural networks to capture the deep features of motion vectors in low-dimensional space remains challenging. Thus, similar to ConvMatch (Zhang and Ma 2024) and DeMatch (Zhang et al. 2024), we initially convert a low-dimensional motion vectors into high-dimensional motion vectors $\{\mathbf{f}_i\}_{i=1}^N \in \mathbb{R}^D$ for input layer (layer 0):

$$^{(0)}\mathbf{f}_i = \mathrm{Up}_0(\mathbf{m}_i), \quad i = 1, \dots, N, \tag{1}$$

where $\mathrm{Up}_0(\cdot)$ means mapping the motion vectors from low-dimensional space to high-dimensional feature space by conducting positional embedding (Vaswani et al. 2017), resulting in more distinctive high-dimensional motion vectors.

Further, we convert the keypoint coordinates from image $\mathbf{I}_A$ into high-dimensional positional vectors $\{\mathbf{p}_i\}_{i=1}^N \in \mathbb{R}^{\frac{D}{2}}$, allowing us to leverage the deep features of the coordinates of initial points to create deep kernels for later interpolation:

$$\mathbf{p}_i = \mathrm{Up}_1(\mathbf{x}_i), \quad i = 1, \dots, N, \tag{2}$$

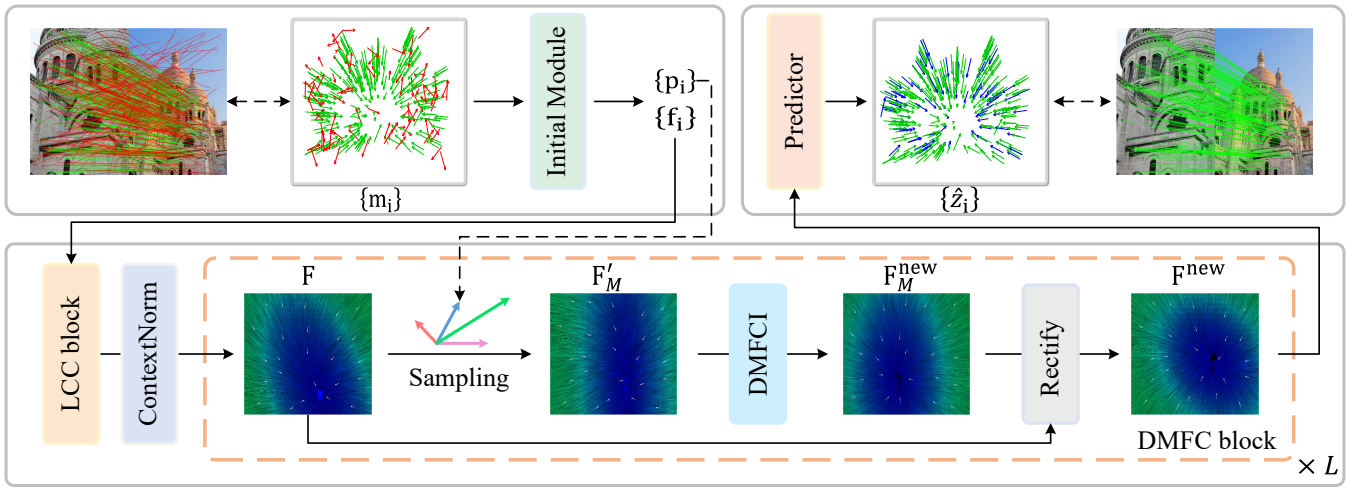

Figure 1: Architecture of DeMo. Given the putative correspondences as input, the Initial Module transforms the motion vectors $\{\mathbf{m}_i\}_{i=1}^N$ and the coordinates of initial points $\{\mathbf{x}_i\}_{i=1}^N$ to $\{\mathbf{f}_i\}_{i=1}^N$ and $\{\mathbf{p}_i\}_{i=1}^N$ in high-dimension space, respectively, where the latter are used to generate the kernels for interpolation. Then, the LCC block performs consensus complementation for feature and spatial channels within the k-nearest neighbors of $\mathbf{f}_i$. After enhancement by ContextNorm, we perform scene-adaptive sampling on $\{\mathbf{f}_i\}_{i=1}^N$ and $\{\mathbf{p}_i\}_{i=1}^N$, and utilize DMFCI for deep motion consensus interpolation. The interpolated new motion is used to smoothly rectify the initial motion field. As the output, we predict the logits $\{\hat{\mathbf{z}}_i\}_{i=1}^N$ for each correspondence, which are used to calculate losses and classify inliers/outliers.

where $\mathrm{Up}_1(\cdot)$ means upgrading the dimension of the input vector to $\frac{D}{2}$.

Deep Motion Field Consensus Block

To generate a robust and representative deep underlying motion field in the case where the putative set contains substantial outliers, we formulate it into an optimization problem under a certain regularization. Kernels are highly flexible in measuring the relationships between components of the motion vector field as well as the relationships with global inputs, making them well-suited for global consensus interpolation in high-dimensional motion fields. Following this train of thought, we incorporate Tikhonov regularization in RKHS with learnable deep kernels to capture the consensus of motion field in high dimensions, which is fleshed out by the Deep Motion Field Consensus (DMFC) block.

Deep Motion Field Consensus Interpolation Given the high-dimensional motion vector $\mathbf{f}_i$ and coordinate $\mathbf{p}_i$ as input to the Deep Motion Field Consensus Interpolation (DMFCI) module, our goal is to interpolate the motion field $\mathbf{g}$ and thereby predict the new motion vector $\mathbf{f}_i^{\text{new}}$ at each location. Concretely, with the input $\{\mathbf{p}_i\}_{i=1}^N$ and a reproducing kernel $\mathbf{\Gamma}$, the unique RKHS $\mathcal{H}$ can be defined as follows:

$$\mathcal{H} = \sum_{i=1}^N \mathbf{\Gamma}(\cdot, \mathbf{p}_i)\mathbf{c}_i, \quad \mathbf{c}_i \in \mathbb{R}^D, \tag{3}$$

where $\mathbf{c}_i$ is a coefficient.

The space norm is given by the following inner product:

$$(\mathbf{g}_1, \mathbf{g}_2)_{\mathcal{H}} = \sum_{i,j=1}^N (\mathbf{\Gamma}(\mathbf{p}_j, \mathbf{p}_i)\mathbf{c}_i, \mathbf{a}_j), \quad \forall \mathbf{g}_1, \mathbf{g}_2 \in \mathcal{H}, \tag{4}$$

where $\mathbf{g}_1 = \sum_{i=1}^N \mathbf{\Gamma}(\cdot, \mathbf{p}_i)\mathbf{c}_i$ and $\mathbf{g}_2 = \sum_{j=1}^N \mathbf{\Gamma}(\cdot, \mathbf{p}_j)\mathbf{a}_j$.

In light of the defined RKHS $\mathcal{H}$, We formulate the deep motion field consensus interpolation problem as follows:

$$\mathcal{E}(\mathbf{g}) = \min_{\mathbf{g}\in\mathcal{H}} \sum_{i=1}^N \|\mathbf{f}_i - \mathbf{g}(\mathbf{p}_i)\|^2 + \lambda\|\mathbf{g}\|_{\mathcal{H}}^2, \tag{5}$$

where $\|\mathbf{f}_i - \mathbf{g}(\mathbf{p}_i)\|^2$ penalizes the error between the predicted and true high-dimensional motion vectors, $\|\mathbf{g}\|_{\mathcal{H}}^2$ is the regularization term used to maintain the smoothness of the learned motion field, and λ is a regularization parameter.

According to the representer theorem (Micchelli and Pontil 2005), the solution of Eq. (5) has the following form:

$$\mathbf{g}(\mathbf{p}) = \sum_{i=1}^N \mathbf{\Gamma}(\mathbf{p}, \mathbf{p}_i)\mathbf{c}_i. \tag{6}$$

The coefficients $\{\mathbf{c}_i | i = 1, 2, \ldots, N\}$ are obtained by solving the following linear system:

$$(\tilde{\mathbf{\Gamma}} + \lambda\mathbf{I})\tilde{\mathbf{C}} = \tilde{\mathbf{F}}, \tag{7}$$

where $\tilde{\mathbf{\Gamma}} \in \mathbb{R}^{DN\times DN}$ is an $N \times N$ block matrix, and $\mathbf{\Gamma}(\mathbf{p}_i, \mathbf{p}_j)$ is the (i, j)-th block of $\tilde{\mathbf{\Gamma}}$. Additionally, $\tilde{\mathbf{F}} = (\mathbf{f}_1^T, \mathbf{f}_2^T, \ldots, \mathbf{f}_N^T)^T$ and $\tilde{\mathbf{C}} = (\mathbf{c}_1^T, \mathbf{c}_2^T, \ldots, \mathbf{c}_N^T)^T$, with $\mathbf{I}$ being the identity matrix.

The choice of kernel, which determines the property of the norm in RKHS, is driven by the complexity of the motion field. We opt for a decomposable kernel, which is adequate for the generated motion field in the context of two-view correspondence learning. Specifically, it takes the following form:

$$\mathbf{\Gamma}(\mathbf{p}, \mathbf{p}') = \kappa(\mathbf{p}, \mathbf{p}')\mathbf{Q}, \tag{8}$$

where $\mathbf{Q}$ is a $D \times D$ positive semi-definite matrix and the scalar kernel κ is defined as the Gaussian kernel:

$$\kappa(\mathbf{p}, \mathbf{p}') = e^{-\beta\|\mathbf{p}-\mathbf{p}'\|^2}. \tag{9}$$

We found that when the positive semi-definite matrix $\mathbf{Q}$, which is used to measure the relationships between output components, is chosen as an identity matrix, the model performance is sufficiently efficient. Therefore, the linear system (7) can be simplified to a more concise linear system:

$$(\mathbf{K} + \lambda \mathbf{I})\mathbf{C} = \mathbf{F}, \tag{10}$$

where $\mathbf{C} = (\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_N)^T$ and $\mathbf{F} = (\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_N)^T$ are both $N \times D$ matrices, and $\mathbf{K} \in \mathbb{R}^{N \times N}$ is a Gram matrix where $\mathbf{K}_{i,j} = \kappa(\mathbf{p}_i, \mathbf{p}_j)$.

Once the kernel function is determined, optimizing Eq. (5) is equivalent to solving Eq. (10). The solution is:

$$\mathbf{f}'_i = \mathbf{g}(\mathbf{p}_i) = \textstyle\sum_{n=1}^{N} \mathbf{K}(\mathbf{p}_i, \mathbf{p}_n)\mathbf{c}_n. \tag{11}$$

Subsequently, Eq. (11) will be used to update the feature vector $\mathbf{f}_i$ at the corresponding position $\mathbf{p}_i$, and the predicted motion vector $\mathbf{f}'_i$ will be used to smoothly rectify the entire deep motion field through the following equation:

$$\mathbf{F}^{\text{new}} = \text{Aggr}(\mathbf{F}, \mathbf{F}'), \tag{12}$$

where $\text{Aggr}(\cdot, \cdot)$ refers to the graph attention network.

In practice, the putative correspondences contain numerous and unevenly distributed outliers, which disrupt the interpolation results, leading to decreased matching accuracy. Therefore, to enhance the network's robustness to outliers, we propose to weight the importance of each motion vector in the process of solving the linear system (10). More concretely, we first predict the inlier weight for each motion vector with an MLP:

$$w_i = MLP(\mathbf{f}_i). \tag{13}$$

Subsequently, we extend the original linear system (10) to a weighted formulation as follows:

$$(\mathbf{WKW} + \lambda \mathbf{I})\mathbf{C} = \mathbf{WF}, \tag{14}$$

where $\mathbf{W} = \text{diag}(w_1, w_2, \ldots, w_N) \in \mathbb{R}^{N \times N}$ is a diagonal matrix. By this means, DeMo can achieve deep global motion consensus interpolation robustly.

Scene-Adaptive Sampling In the process of DMFCI, we need to solve Eq. (14), which is both theoretically and experimentally feasible but has a daunting computational complexity of $O(N^3)$, thus discouraging real-time visual tasks. Here, we introduce a scene-adaptive sampling strategy that aims to drastically reduce computational complexity by implicitly sampling the more scene-representative motions.

Rather than employing a hard sampling, each motion participates in the subsequent interpolation in a soft sampling manner based on its own redundancy of scene information. Proceeding from this point, we first use an MLP to generate a weight matrix $\mathbf{W}' \in \mathbb{R}^{M \times N}$ from $\mathbf{F}$. The operation can be implemented using the following matrix multiplications:

$$\mathbf{F}_M = \mathbf{W}'\mathbf{F}, \tag{15}$$

$$\mathbf{P}_M = \mathbf{W}'\mathbf{P}, \tag{16}$$

where $\mathbf{P} = (\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_N)^T$ is an $N \times \frac{D}{2}$ matrix. To enhance the details and edges (i.e., low-frequency) of the motion field on the representation motion, we inject the motion vectors $\mathbf{F}$ into the scene-representative ones as follows:

$$\hat{\mathbf{F}}_M = \text{Aggr}(\mathbf{F}_M, \mathbf{F}). \tag{17}$$

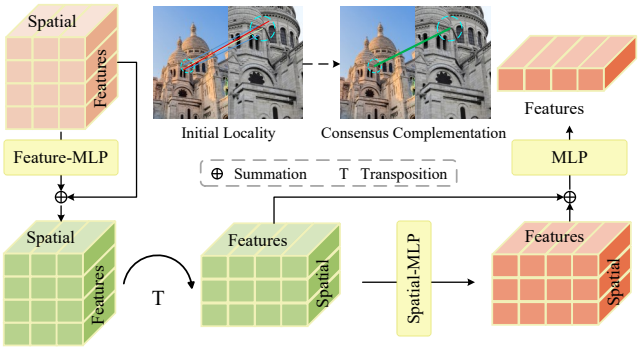

Figure 2: Architecture details of LCC block.

After ensuring that the representative information of the original motion field is fully preserved, we compute the Gram matrix $\mathbf{K}_M \in \mathbb{R}^{M \times M}$ in Eq. (14) of DMFCI with $\mathbf{P}_M$ and the Gaussian kernel function. Consequently, $\hat{\mathbf{F}}_M$ can also be fed into the DMFCI for interpolation to obtain $\mathbf{F}_M^{\text{new}}$, which are used to smoothly rectify the initial motion field:

$$\mathbf{F}^{\text{new}} = \text{Aggr}(\mathbf{F}, \mathbf{F}_M^{\text{new}}). \tag{18}$$

The computational complexity of the $\text{Aggr}(\cdot, \cdot)$ operation in Eq. (17) and the linear system in Eq. (14) together is $O(MN + M^3)$. Due to $M \ll N$, the time complexity can be written as $O(MN)$. This strategy provides a new perspective for efficiently processing large-scale datasets.

Local Consensus Complementation Block

Considering the underlying depth discontinuities in scenes, we propose to pre-process the high-dimensional motion field with an innovative LCC block that uses the spatial channels as the consensus complement to the feature ones, before implementing global motion consensus interpolation. Through the sequential implementation of consensus complement operations in both channels, DeMo is able to capture local motion consensus and identify changes in the motion field brought about by differences in depth maps.

Prior to local operations, we employ a bottleneck structure to reduce computational redundancy in the integration process of all local context, by reducing the motion vector dimensions to D_1 to obtain ${}^{(\ell)}\mathbf{f}'_i \in \mathbb{R}^{D_1}$ at the ℓ-th layer. This is because each local motion pattern is relatively simple, making extensive computation and complex designs unnecessary. Following this step, we construct a neighborhood for each motion vector with k-nearest neighbor search. Given the i-th motion vector ${}^{(\ell)}\mathbf{f}'_i$, we compute the differences between the motion vector and its neighbors ${}^{(\ell)}\mathbf{f}'_{i,j}$ by ${}^{(\ell)}\mathbf{e}_{i,j} = {}^{(\ell)}\mathbf{f}'_i - {}^{(\ell)}\mathbf{f}'_{i,j}$, where $j \in \mathbb{N}_{K_0}$.

As shown in Figure 2, the LCC block first employs an MLP to capture the consensus of the local feature channels and prevents information loss using skip-connection. Next, it uses another MLP to process the spatial channels of each neighborhood, similarly incorporating skip-connection. Finally, an additional MLP is utilized to integrate all local con-

text. These processes can be summarized as follows:

$$^{(\ell)}\mathbf{S}_i = MLP(\,^{(\ell)}\mathbf{S}'_i) + \,^{(\ell)}\mathbf{S}'_i, \tag{19}$$

$$^{(\ell+1)}\mathbf{f}'_i = MLP(MLP(\,^{(\ell)}\mathbf{S}^T_i) + \,^{(\ell)}\mathbf{S}^T_i), \tag{20}$$

where $^{(\ell)}\mathbf{S}'_i = (\,^{(\ell)}\mathbf{e}_{i,1}, \,^{(\ell)}\mathbf{e}_{i,2}, \ldots, \,^{(\ell)}\mathbf{e}_{i,K_0})$.

Once local consensus complementation is completed, we raise the dimension back to D. Subsequently, we use a ContextNorm operation to further enhance context, thereby mitigating the feature space discrepancies between local and global modules caused by their separate processing.

Loss Functions

The loss function is a linear combination of classification and regression loss functions (Zhang et al. 2019):

$$\mathcal{L} = \sum_{0\le\ell\le\frac{L}{2}-1} \mathcal{L}_{cls}\left(\mathbf{z}, \,^{(2\ell+1)}\hat{\mathbf{z}}\right) + \mu\mathcal{L}_{reg}\left(\mathbf{E}, \,^{(2\ell+1)}\hat{\mathbf{E}}\right), \tag{21}$$

where $2\ell+1$ denotes the odd-numbered layers in the network, and μ is a hyper-parameter to balance them.

The classification loss $\mathcal{L}_{cls}$ is calculated using the binary cross-entropy loss between the predicted probability $^{(2\ell+1)}\hat{\mathbf{z}}$ at the odd layers and the weak supervision label $\mathbf{z}$, which is generated by calculating the Sampson distance with a threshold of 10^{-4}. The regression loss $\mathcal{L}_{reg}$ is also derived from the Sampson distance, with the loss defined as:

$$\mathcal{L}_{reg}\left(\mathbf{E}, \hat{\mathbf{E}}\right) = \sum_{i=1}^{N} \frac{(\mathbf{v}_i^T\hat{\mathbf{E}}\mathbf{u}_i)^2}{\|\mathbf{E}\mathbf{u}_i\|^2_{[1]} + \|\mathbf{E}\mathbf{u}_i\|^2_{[2]} + \|\mathbf{E}^T\mathbf{v}_i\|^2_{[1]} + \|\mathbf{E}^T\mathbf{v}_i\|^2_{[2]}}, \tag{22}$$

where $\mathbf{v}_i$ and $\mathbf{u}_i$ denote homogeneous coordinates of two keypoints which form a virtual correspondence generated based on the ground truth essential matrix $\mathbf{E}$, while $\hat{\mathbf{E}}$ is the essential matrix estimated at odd-numbered layers using the weighted eight-point algorithm (Longuet-Higgins 1981). $\|\mathbf{u}\|_{[m]}$ denotes the m-th element of the vector $\mathbf{u}$.

Implementation Details

DeMo consists of eight DMFC layers (i.e., $L = 8$), the channel dimension D is 128, and the scene-adaptive sampling parameter M is set to 48. In the LCC block, the number of neighbors within a single neighborhood and the bottleneck dimension are both set to 8. We establish up to $N = 2000$ putative matches with SIFT for each image pair. Notably, both the regularization parameter λ and the parameter β in the Gaussian kernel are trainable. To ensure stability during numerical computation, the values of weight matrix $\mathbf{W}$ in Eq. (14) are constrained to the range $[0.05, 0.95]$. When training with the Adam optimizer, the learning rate is set to 10^{-4} for the first $80k$ iterations and then gradually decreases at a rate of 0.999996 every step. The batch size for training is 32. In the loss function, parameter μ is set to 0 for the first $20k$ iterations and then adjusted to 0.5 for the remaining iterations. For outdoor and indoor scenes, DeMo terminates training at $500k$ and $700k$ iterations, respectively. All training and testing are conducted on a single RTX3090 GPU.

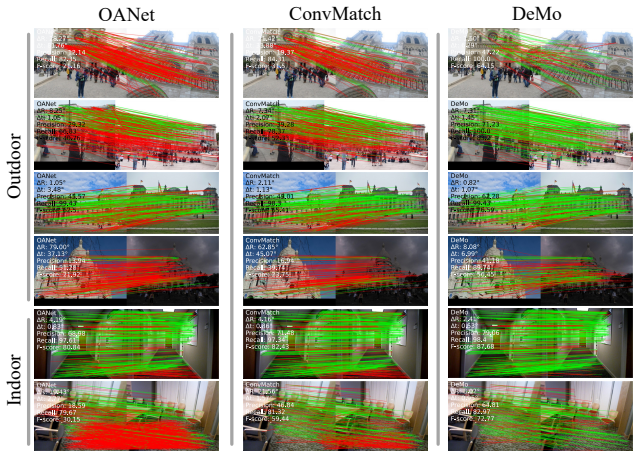

Figure 3: Qualitative illustration of outlier rejection. The false matches are marked in red while correct matches are marked in green. The results of relative pose estimation and outlier rejection are provided in the top left corners.

Experiments

Relative Pose Estimation

Relative pose estimation refers to using the inliers predicted from the image pair to estimate the relative position between cameras, including rotation and translation.

Datasets For outdoor scene, YFCC100M (Thomee et al. 2016) consists of 100 million images from Internet, divided into 72 sequences. Following the setup of OANet (Zhang et al. 2019), we select 68 sequences for training and validation, leaving 4 unseen sequences for testing. For indoor scene, SUN3D (Xiao, Owens, and Torralba 2013) comprises indoor images sampled at every 10 frames from RGBD videos. Analogously, we select 239 sequences for training and validation, reserving 15 sequences for testing. In our experiments, we detect up to 2000 keypoints on each image using SIFT and establish the putative set with the NN method.

Evaluation Protocols We report the Area Under the Curve (AUC) of the cumulative error curves at multiple thresholds $(5^\circ, 10^\circ, 20^\circ)$, where the pose error is the maximum angular error during rotation and translation. In the experiment, we utilize the weighted eight-point algorithm and RANSAC (Fischler and Bolles 1981) to estimate the essential matrix, thereby recovering the relative poses.

Baselines DeMo is compared against traditional handcrafted methods such as CRC (Fan et al. 2023), GMS (Bian et al. 2017), LPM (Ma et al. 2019), and VFC (Ma et al. 2014), as well as learning-based methods like PointCN (Yi et al. 2018), OANet (Zhang et al. 2019), LMCNet (Liu et al. 2021), NCMNet (Liu and Yang 2023), ConvMatch (Zhang and Ma 2024), UMatch (Li, Zhang, and Ma 2023), and DeMatch (Zhang et al. 2024).

Results of Relative Pose Estimation All quantitative results are shown in Table 1. For outdoor scenes, DeMo significantly surpasses other methods across all error thresh-

Method	YFCC100M			SUN3D		
	@5°	@10°	@20°	@5°	@10°	@20°
CRC	- / 12.05	- / 23.17	- / 36.53	- / 4.07	- / 10.44	- / 20.82
GMS	- / 13.29	- / 24.38	- / 37.83	- / 4.12	- / 10.53	- / 20.82
LPM	- / 15.99	- / 28.25	- / 41.76	- / 4.80	- / 12.28	- / 23.77
VFC	- / 17.43	- / 29.98	- / 43.00	- / 5.26	- / 13.05	- / 24.84
PointCN	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	15.92 / 27.26	35.93 / 45.93	57.11 / 63.17	5.93 / 6.78	16.91 / 17.10	34.32 / 32.41
LMCNet	22.35 / 30.48	43.57 / 49.84	63.34 / 66.94	7.08 / 6.84	19.09 / 17.62	37.15 / 33.43
NCMNet	26.89 / 32.30	46.19 / 52.29	64.21 / 69.65	6.31 / 7.10	16.84 / 18.56	33.11 / **35.58**
ConvMatch	29.43 / 33.29	51.27 / 52.96	69.58 / 69.82	9.08 / 7.37	22.88 / 18.69	41.27 / 34.85
UMatch	30.93 / 33.92	52.17 / 53.09	69.75 / 69.45	8.03 / 7.01	20.83 / 17.79	38.75 / 33.57
DeMatch	30.89 / 32.98	52.67 / 52.37	70.33 / 69.01	**9.31** / 7.44	23.10 / 18.66	41.55 / 34.78
DeMo	**32.57 / 34.63**	**54.82 / 54.42**	**72.51 / 71.15**	**9.31 / 7.48**	**23.18 / 18.85**	**41.75** / 35.31

Table 1: Quantitative results of relative pose estimation with weighted eight-point / RANSAC. AUC at $5°$, $10°$, $20°$ is reported.

Method	HPatches			F.	Time(ms)
	Acc.@3px	Acc.@5px	Acc.@10px		
PointCN	38.97 / 68.97	51.38 / 83.10	65.52 / 92.59	81.74	**9.33**
OANet	39.83 / 68.62	52.41 / 82.59	63.10 / 91.90	81.85	14.52
LMCNet	47.93 / 72.24	58.62 / 85.34	69.83 / 92.59	85.45	372.48
ConvMatch	46.72 / 71.03	58.62 / 83.79	68.28 / 91.90	81.95	53.60
UMatch	47.07 / 70.34	57.41 / 85.00	70.00 / 91.72	82.95	84.43
DeMatch	46.90 / 70.69	60.52 / 82.76	71.55 / **92.59**	83.11	56.27
DeMo	**53.79 / 73.10**	**63.79 / 85.52**	**75.69 / 92.59**	**86.13**	64.46

Table 2: Evaluation of homography estimation. Accuracy (Acc.) of estimated homographies at different error thresholds (with DLT / RANSAC post-processing) is reported.

Method	Day	Night
	(0.25m, 2°) / (0.5m, 5°) / (5.0m, 10°)	
PointCN	83.1 / 92.2 / 96.2	69.4 / 79.6 / 89.8
OANet	83.1 / 92.5 / 96.6	72.4 / 80.6 / 90.8
LMCNet	84.1 / 92.8 / 97.1	71.4 / 81.6 / 93.9
ConvMatch	84.6 / 92.7 / **97.2**	72.4 / 83.7 / 91.8
UMatch	85.3 / 92.6 / 96.8	72.4 / 82.7 / 90.8
DeMatch	85.2 / 92.8 / 97.1	73.5 / **84.7 / 94.9**
DeMo	**85.8 / 93.1** / 96.8	**74.5 / 84.7 / 94.9**

Table 3: Evaluation results of Visual localization.

olds, whether utilizing the weighted eight-point algorithm or RANSAC. DeMo also shows superior performance on indoor scenes. We further present the qualitative results in Figure 3. Such outstanding performance reveals that capturing the consensus of the entire motion field allows for accurate identification of the overall spatial transformation between the image pair. This distinctive characteristic of DeMo enables it to perform much more satisfactorily under both large viewpoint changes and illumination variations.

Homography Estimation

Dataset We conduct homography estimation on the HPatches benchmark (Balntas et al. 2017), which includes 116 scenes with a total of 696 images. Among these, 57 scenes are captured under different illumination conditions, while the remaining scenes underwent changes in viewpoint. Each scene consists of one reference image and five target images, with ground-truth homographies provided. We detect up to 4000 keypoints with SIFT.

Evaluation Protocols We adopt homography error at various thresholds ($3/5/10$ pixels) to classify the correctness of the estimates and compute the average accuracy (**Acc.**) across all images. Additionally, we report the F-score (**F.**) for inlier/outlier classification and the computation time per image. Notably, we evaluate all methods with robust RANSAC and non-robust Direct Linear Transform (DLT), respectively.

Results As shown in Table 2, DeMo outperforms other baselines by a significant margin, both in terms of accuracy at the majority of thresholds and the outlier rejection criterion F-score. In particular, the computation time of DeMo is close to that of DeMatch, which employs implicit regularization to lower explicit computation costs.

Visual Localization

The objective of visual localization is to estimate the 6-degree of freedom camera pose (6-DOF) of a given reference image with respect to its 3D scene model, which requires robust and accurate matching algorithms to account for changes in illumination and large viewpoint variations.

Dataset We integrate DeMo into the HLoc (Sarlin et al. 2019) pipeline and evaluate it on the Aachen Day-Night benchmark (Sattler et al. 2018), which consists of $4,328$ reference images of Aachen City and 922 query images, including 824 day-time images and 98 night-time images taken by mobile phone cameras.

Evaluation Protocols In particular, We extract up to $4,096$ keypoints per image with SIFT, and establish putative matches with the mutual nearest neighbor (MNN) method. Following this, we triangulate an SfM model from day-time images with known poses and register both day-time and night-time query images using 2D-2D matches and COLMAP (Schonberger and Frahm 2016). The percentage of correctly localized queries is reported based on specific distance and rotation thresholds.

Results Results are reported in Table 3. DeMo performances better overall, particularly showing outstanding results on the more challenging night-time scenes.

Method	YFCC100M			SUN3D			
	RootSIFT	LIFT	SuperPoint	SIFT	RootSIFT	LIFT	SuperPoint
PointCN	24.71	15.31	14.94	1.56	1.71	1.99	3.52
OANet	36.46	28.83	20.54	3.37	3.59	3.20	3.39
ConvMatch	52.01	42.89	**31.14**	7.26	7.57	6.56	**5.55**
UMatch	52.49	38.86	25.90	7.63	7.66	4.84	3.45
DeMatch	53.32	42.27	29.84	7.12	7.31	6.35	5.18
DeMo	**55.56**	**44.23**	28.45	**8.22**	**8.53**	**6.89**	5.01

Table 4: Generalization ability test. AUC@10° with the weighted eight-point algorithm is reported.

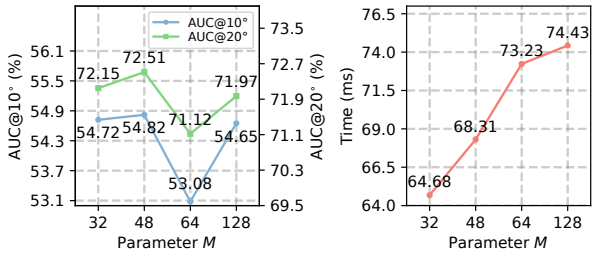

Figure 4: Parameter setting. We repeat relative pose estimate on YFCC100M with SIFT while changing the hyperparameter M. AUC@10°, AUC@20°, and computation time for each image pair without RANSAC are reported.

Analysis

This section analyzes DeMo's generalization across descriptors and datasets, sampling hyperparameters, and includes ablation studies to assess its effectiveness.

Generalization Ability In this experiment, all methods are trained on the YFCC100M with SIFT. Subsequently, we evaluate the performance with RootSIFT (Arandjelović and Zisserman 2012), LIFT (Yi et al. 2016), and SuperPoint descriptors on YFCC100M, and SIFT, RootSIFT, LIFT, and SuperPoint descriptors on SUN3D. We extract up to 2000 keypoints for SIFT, RootSIFT, and LIFT, and 1000 keypoints for SuperPoint. As shown in Table 4, DeMo achieves outstanding performance, validating the strong robustness of the motion field consensus prior.

Parameter Setting In DeMo, the scene-adaptive sampling hyperparameter M has a large impact on the model performance. Increasing M increases computation but retains motions with high information redundancy, while decreasing M improves efficiency but degrades performance. In order to trade off performance and efficiency, we test the effect of different M on the outdoor dataset. The results are shown in Figure 4. We select $M = 48$ as the default parameter for balancing performance and efficiency.

Ablation Studies The ablation results on YFCC100M are shown in Table 5. Removing the DMFC block (row iii) significantly degrades performance, demonstrating the motion consensus prior's effectiveness in high-dimensional motion fields for two-view correspondence learning. Furthermore, row (iv) in Table 5 indicates that the interpolation technique with learnable kernels in RKHS enhances motion consensus. To demonstrate the effectiveness of our scene-adaptive sampling strategy, we replace it with the point selection strategy based on inlier scores from the previous layer of the network, and random selection strategy. The results are presented in rows (v) and (vi) of Table 5. Row (vii) is the full DeMo with the scene-adaptive sampling strategy.

Method	YFCC100M		
	AUC@5°	AUC@10°	AUC@20°
(i) w/o. LCC block	27.98	49.61	68.26
(ii) w/o. Spatial Channels	31.49	53.78	71.76
(iii) w/o. DMFC block	11.03	27.21	47.52
(iv) w/o. DMFCI	31.78	53.65	71.29
(v) w. Probability-based Selection	30.26	52.15	70.28
(vi) w. Random Selection	30.31	52.28	70.42
(vii) DeMo(full)	**32.57**	**54.82**	**72.51**

Table 5: Ablation studies of DeMo (without RANSAC post-processing) on YFCC100M dataset.

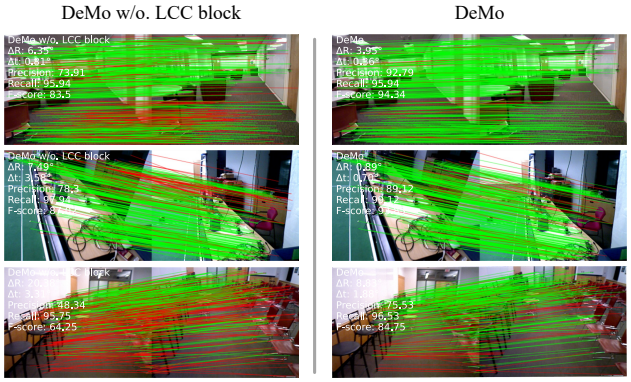

Figure 5: Comparison between DeMo w/o. LCC block and full DeMo. The image pairs involve large depth differences.

Additionally, we remove spatial channel processing and the LCC block separately (Table 5, rows ii and i). To verify that the consensus complementation of feature and spatial channels can mitigate the adverse effects of discontinuities in the motion field, we evaluate the relative pose estimation on SUN3D and illustrate the qualitative results with and without the LCC block in Figure 5. The version with the LCC block reduces mismatches on the ground (the 1st row), the table and cabinet (the 2nd row), and chairs (the 3rd row).

Conclusion

In this paper, we propose a novel network called DeMo for two-view correspondence learning. Drawing upon the long-range prior, DeMo incorporates regularization with learnable kernels in RKHS to capture global motion consensus of the high-dimensional motion field as a way to rectify the motion field and reject outliers. A scene-adaptive sampling strategy is designed to reduce computational complexity by selecting representative motions. To mitigate the effect of motion field discontinuities, we further design an LCC block to capture local consensus while complementing consensus between spatial and feature channels to identify depth differences and avoid over-smoothing. A series of experiments on the mainstream benchmarks consistently demonstrate the superiority of DeMo over the current state-of-the-arts.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (62276192 and U23B2050), and the Fund of National Key Laboratory of Multispectral Information Intelligent Processing Technology (61421132302).

References

Arandjelović, R.; and Zisserman, A. 2012. Three things everyone should know to improve object retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2911–2918.

Aronszajn, N. 1950. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3): 337–404.

Baldassarre, L.; Rosasco, L.; Barla, A.; and Verri, A. 2012. Multi-output learning via spectral filtering. *Machine Learning*, 87: 259–301.

Balntas, V.; Lenc, K.; Vedaldi, A.; and Mikolajczyk, K. 2017. HPatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5173–5182.

Bian, J.; Lin, W.-Y.; Matsushita, Y.; Yeung, S.-K.; Nguyen, T.-D.; and Cheng, M.-M. 2017. GMS: Grid-based motion statistics for fast, ultra-robust feature correspondence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4181–4190.

DeTone, D.; Malisiewicz, T.; and Rabinovich, A. 2018. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 224–236.

Fan, A.; Jiang, X.; Ma, Y.; Mei, X.; and Ma, J. 2023. Smoothness-driven consensus based on compact representation for robust feature matching. *IEEE Transactions on Neural Networks and Learning Systems*, 34(8): 4460–4472.

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

Li, Z.; Zhang, S.; and Ma, J. 2023. U-Match: Two-view Correspondence Learning with Hierarchy-aware Local Context Aggregation. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 1169–1176.

Liu, X.; and Yang, J. 2023. Progressive neighbor consistency mining for correspondence pruning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 9527–9537.

Liu, Y.; Liu, L.; Lin, C.; Dong, Z.; and Wang, W. 2021. Learnable motion coherence for correspondence pruning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3237–3246.

Longuet-Higgins, H. C. 1981. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293(5828): 133–135.

Lowe, D. G. 2004. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60: 91–110.

Ma, J.; Zhao, J.; Jiang, J.; Zhou, H.; and Guo, X. 2019. Locality preserving matching. *International Journal of Computer Vision*, 127: 512–531.

Ma, J.; Zhao, J.; Tian, J.; Yuille, A. L.; and Tu, Z. 2014. Robust point matching via vector field consensus. *IEEE Transactions on Image Processing*, 23(4): 1706–1721.

Micchelli, C. A.; and Pontil, M. 2005. On learning vector-valued functions. *Neural Computation*, 17(1): 177–204.

Mur-Artal, R.; Montiel, J. M. M.; and Tardos, J. D. 2015. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5): 1147–1163.

Philbin, J.; Isard, M.; Sivic, J.; and Zisserman, A. 2010. Descriptor learning for efficient retrieval. In *Proceedings of the European Conference on Computer Vision*, 677–691.

Sarlin, P.-E.; Cadena, C.; Siegwart, R.; and Dymczyk, M. 2019. From coarse to fine: Robust hierarchical localization at large scale. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 12716–12725.

Sattler, T.; Maddern, W.; Toft, C.; Torii, A.; Hammarstrand, L.; Stenborg, E.; Safari, D.; Okutomi, M.; Pollefeys, M.; Sivic, J.; et al. 2018. Benchmarking 6dof outdoor visual localization in changing conditions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8601–8610.

Schonberger, J. L.; and Frahm, J.-M. 2016. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4104–4113.

Thomee, B.; Shamma, D. A.; Friedland, G.; Elizalde, B.; Ni, K.; Poland, D.; Borth, D.; and Li, L.-J. 2016. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59(2): 64–73.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

Xiao, J.; Owens, A.; and Torralba, A. 2013. Sun3d: A database of big spaces reconstructed using sfm and object labels. In *Proceedings of the IEEE International Conference on Computer Vision*, 1625–1632.

Xu, H.; Ma, J.; Yuan, J.; Le, Z.; and Liu, W. 2022. Rfnet: Unsupervised network for mutually reinforcing multi-modal image registration and fusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 19679–19688.

Yi, K. M.; Trulls, E.; Lepetit, V.; and Fua, P. 2016. Lift: Learned invariant feature transform. In *Proceedings of the European Conference on Computer Vision*, 467–483.

Yi, K. M.; Trulls, E.; Ono, Y.; Lepetit, V.; Salzmann, M.; and Fua, P. 2018. Learning to find good correspondences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2666–2674.

Zhang, J.; Sun, D.; Luo, Z.; Yao, A.; Zhou, L.; Shen, T.; Chen, Y.; Quan, L.; and Liao, H. 2019. Learning two-view

correspondences and geometry using order-aware network. In *Proceedings of the IEEE International Conference on Computer Vision*, 5845–5854.

Zhang, S.; Li, Z.; Gao, Y.; and Ma, J. 2024. DeMatch: Deep Decomposition of Motion Field for Two-View Correspondence Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 20278–20287.

Zhang, S.; and Ma, J. 2024. Convmatch: Rethinking network design for two-view correspondence learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5): 2920–2935.

Zhao, C.; Ge, Y.; Zhu, F.; Zhao, R.; Li, H.; and Salzmann, M. 2021. Progressive correspondence pruning by consensus learning. In *Proceedings of the IEEE International Conference on Computer Vision*, 6464–6473.