PMatch: Paired Masked Image Modeling for Dense Geometric Matching

Shengjie Zhu and Xiaoming Liu
Department of Computer Science and Engineering,
Michigan State University, East Lansing, MI, 48824
zhusheng@msu.edu, liuxm@cse.msu.edu

Abstract

Dense geometric matching determines the dense pixel-wise correspondence between a source and support image corresponding to the same 3D structure. Prior works employ an encoder of transformer blocks to correlate the two-frame features. However, existing monocular pretraining tasks, e.g., image classification, and masked image modeling (MIM), can not pretrain the cross-frame module, yielding less optimal performance. To resolve this, we reformulate the MIM from reconstructing a single masked image to reconstructing a pair of masked images, enabling the pretraining of transformer module. Additionally, we incorporate a decoder into pretraining for improved upsampling results. Further, to be robust to the textureless area, we propose a novel cross-frame global matching module (CFGM). Since the most textureless area is planar surfaces, we propose a homography loss to further regularize its learning. Combined together, we achieve the State-of-The-Art (SoTA) performance on geometric matching. Codes and models are available at https://github.com/ShngJZ/PMatch.

1. Introduction

When a 3D structure is viewed in both a source and a support image, for a pixel (or keypoint) in the source image, the task of geometric matching identifies its corresponding pixel in the support image. This task is a cornerstone for many downstream vision applications, *e.g.* homography estimation [18], structure-from-motion [45], visual odometry estimation [21] and visual camera localization [7].

There exist both sparse and dense methods for geometric matching. The sparse methods [16, 19, 32, 33, 40, 42, 48, 48, 56] only yield correspondence on sparse or semi-dense locations while the dense methods [20, 54, 55] estimate pixel-wise correspondence. They primarily differ in that the sparse methods embed a keypoint detection or a global matching on discrete coordinates, which underlyingly assumes a unique mapping between source and support frames. Yet, the existence of textureless surfaces introduces multiple similar local patches, disabling keypoint detection or causing ambiguous matching results. Dense methods, though facing similar challenges at the coarse level, alleviate it with the additional fine-level local context and smoothness constraint. Until recently, the dense methods demonstrate a comparable or better geometric matching performance over the sparse methods [20, 54, 55].

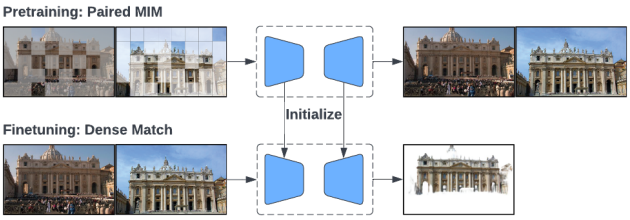


Figure 1. Most vision tasks start with a pretrained network. In geometric matching, the unique network components processing two-view features cannot benefit from the monocular pretraining task, *e.g.*, image classification, and masked image modeling (MIM). As in the figure, this work enables the pretraining of a matching model via reformulating MIM from reconstructing a single masked image to reconstructing a pair of masked images.

A relevant task to dense geometric matching is the optical flow estimation [50]. Both tasks estimate dense correspondences, whereas the optical flow is applied over consecutive frames with the constant brightness assumption.

In geometric matching [9, 48], apart from the encoder encodes source and support frames into feature maps, there exist transformer blocks which correlate two-frame features, *e.g.*, the LoFTR module [48]. Since these network components consume two-frame inputs, the monocular pretraining task, *e.g.*, the image classification and masked image modeling (MIM) defined on ImageNet dataset, is unable to benefit the network. This limits both the geometric matching performance and its generalization capability.

To address this, we reformulate the MIM from single masked image reconstruction to paired masked images reconstruction, *i.e.*, pMIM. Paired MIM benefits the geometric matching as both tasks rely on the cross-frame module to correlate two frames inputs for prediction.

With a pretrained encoder, the decoder in dense geometric matching is still randomly initialized. Following the idea of pretraining encoder, we extend pMIM pretraining to the decoder. As part functionality of decoder is to upsample the coarse-scale initial prediction to the same resolution as input, we also task the decoder in pMIM to upsample the coarse-scale reconstruction to its original resolution. Correspondingly, we consist the decoder as stacks of the depth-wise convolution except for the last prediction head. With the depth-wise decoder, when transferring from pMIM to geometric matching, we duplicate the decoder along the channel dimension to finish the initialization. To this end, there exists only a small number of components in the decoder randomly initialized, we pretrain the rest network components using synthetic image pair augmentation [54].

To further improve the dense geometric matching performance, we propose a cross-frame global matching module (CFGM). In CFGM, we first compute the correlation volume. We model the correspondences of coarse scale pixels as a summation over the discrete coordinates in the support frame, weighted by the softmaxed correlation vector. However, this modeling fails when multiple similar local patches exit. As a solution, we impose positional embeddings to the discrete coordinates and decode with a deep architecture to avoid ambiguity. Meanwhile, we notice that the textureless surfaces are mostly planar structures described by a low-dimensional 8 degree-of-freedom (DoF) homography matrix. We thus design a homography loss to augment the learning of the low DoF planar prior.

We summarize our contributions as follows:

• We introduce the paired masked image modeling pretext task, pretraining both the encoder and decoder of a dense geometric matching network.

• We propose a novel cross-frame global matching module that is robust to textureless local patches. Since the most textureless patches are planar structures, we augment their learning with a homography loss.

• We outperform dense and sparse geometric matching methods on diverse datasets.

2. Related works

2.1. Pretraining and Finetuning

Pretraining and finetuning is an effective paradigm in vision tasks. Supervised image classification has been one of the most widely adopted pretraining methods. An encoder [24, 25, 47], *e.g.*, ResNet [24], together with a few fully connected (FC) layers is trained for image classification using a large-scale dataset, *e.g.*, ImageNet [14]. After converging, the encoder is used as the initialization in the downstream vision tasks.

Apart from supervised classification tasks, there are self-supervised methods producing discriminative feature representation. Inspired by BYOL [22], DINO [8] introduces a self-supervised mean-teacher knowledge distillation task. It encourages the prediction consistency between a student and teacher model where the teacher is an exponential moving average of the student model. The pretrained ViT model embeds explicit information of semantic segmentation, which is not observed in a supervised counterpart. Other self-supervised pretraining methods include color transformation [11], geometric transformation [11], Jigsaw Puzzle [35], feature frame prediction [39], *etc*.

Among the self-supervised learning tasks, masked image modeling (MIM) [3, 23, 58, 62, 64, 68] achieves SoTA finetuning performance on ImageNet [14]. The task introduces Masked Language Modeling used in NLP domain to vision, reconstructing an image from its masked input. While iGPT [10], ViT [17], and BEiT [3] adopt sophisticated paradigm in modeling, MAE [23] and SimMIM [63] show that directly regressing the masked continuous RGB pixels can achieve competitive results. Typically, they focus on pretraining the encoder, adopting an asymmetric design where only a shallow decoder head is appended.

In this paper, we reformulate MIM from reconstructing a single image to the paired images, reducing the domain gap between the pretexting task and the downstream geometric matching. As a result, we extend the benefit of MIM pretraining to the task of dense geometric matching.

2.2. Sparse Geometric Matching

There are detector-based and detector-free sparse geometric matching methods. Classic works are detector based, and employ the nearest neighbor (NN) match using the hand-crafted feature on detected keypoints, *e.g.*, SIFT [33], SURF [5], and ORB [43]. Both keypoint detection and feature extraction are improved by data-driven deep models [16, 16, 19, 38, 40, 66]. Later, [42, 44, 56] propose to replace the naive NN match by graph neural network based differentiable matching.

While the detector based methods operate on keypoints, the detector free methods, *e.g.* LoFTR [48] and ASpanFormer [9] operate all-to-all matching on coarse-scale discrete grid locations. Still, their matching depends on the correlation between features, yielding ambiguous results when multiple local patches exist. We improve LoFTR from two perspectives. First, we extend the LoFTR module to the proposed cross-frame global matching module to benefit from the MIM pretexting task. Second, we alleviate the ambiguity caused by similar local patches by imposing positional embeddings over the low-dimensional 2D coordinates. A decoder is then employed to resolve the ambiguity.

2.3. Dense Geometric Matching

DGC-Net [34] regresses dense correspondences from a global correlation volume at a limited resolution. GLU-

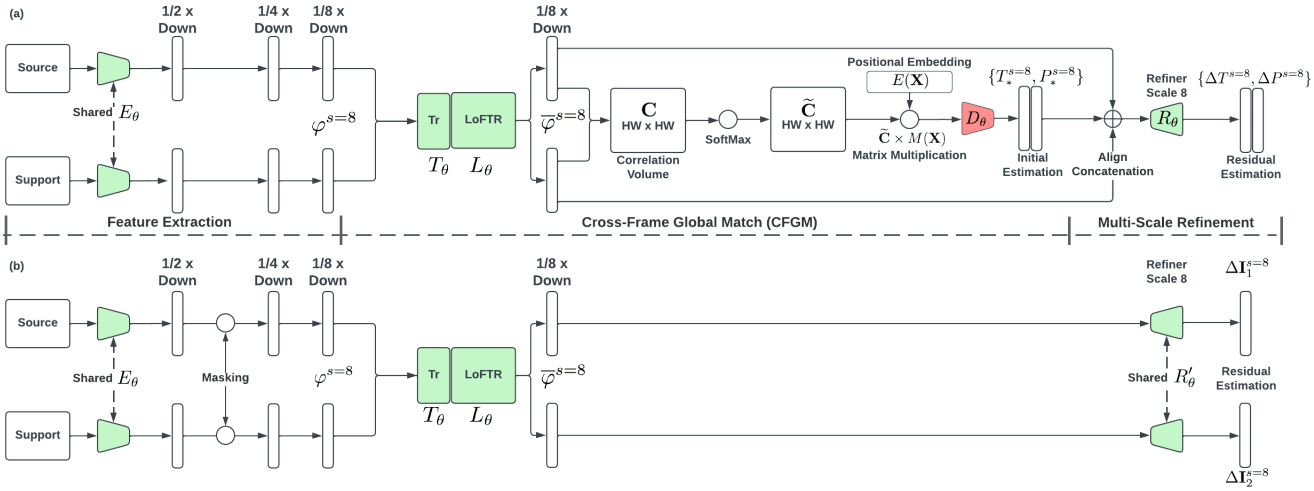


Figure 2. **Methodology Overview.** In (a), we illustrate the proposed dense geometric matching network. After extracting the multi-scale feature with the encoder E_θ, we extend the LoFTR module with (1) Transformer blocks T_θ and (2) positional embeddings with an appended decoder D_θ to remove the ambiguity when multiple local patches exist. In (b), we show the proposed paired MIM pretext task. We apply image masking at the scale $s = 2$, and recover the masked images with the transformer blocks. In (a), network D_θ (in red) is not included in pMIM pretraining. In dense matching, R_θ takes in the stack of source and the aligned support frame feature. In the pretext task, R'_θ only takes in the source frame feature. Thus, R'_θ is a sub-graph of R_θ. We detail how to initialize R_θ using R'_θ in Fig. 3. The residual refinement at other scales repeats the process at scale $s = 8$ but consumes feature embeddings of other scales, skipped for simplicity.

Net [53] increases the resolution with a global-local correlation layer. GOCor [52] further improves GLU-Net [53] by replacing the correlation layer with online optimization. Other methods, such as RANSAC Flow [46], iteratively recover a homography transformation to reduce the visual difference between the source and support images.

Though dense methods estimate more correspondences than sparse methods, it is less favored for geometric matching. Until recently, PDC Net+ [54] and DKM [20] close the gap between dense and sparse methods. Both methods model the dense match as probability functions. PDC Net+ adopts a mixture Laplacian distribution while DKM models with the Gaussian Process (GP). Furthermore, they estimate a confidence score to remove false positive results. We follow [20, 54] in the confidence estimation. However, instead of applying probabilistic regression, we keep the correlation based explicit matching process. This saves the computation of the inverse matrix required in the GP Regression of DKM. Also, we apply a unique architecture design to benefit from the MIM pretexting task.

3. Method

In this section, we first introduce the proposed dense geometric matching method. Then we discuss how to pretext the network via the paired masked image modeling. Fig. 2 depicts our framework in finetuning and pretexting stages.

3.1. Dense Geometric Matching

Dense geometric matching computes the dense correspondences between the source image $\mathbf{I}_1$ and support image $\mathbf{I}_2$. Under the estimated correspondences T, source image $\mathbf{I}_1$ can be recovered from support image $\mathbf{I}_2$ by applying bilinear sampling at T. Since the dense correspondences between $\mathbf{I}_1$ and $\mathbf{I}_2$ is not guaranteed to exist at each pixel location, we follow [20] in estimating confidence P to indicate the fidelity of the prediction.

Feature Extraction. As shown in Fig. 2, we adopt a multi-scale ResNet-based [24] feature extractor E_θ. Taking the source frame $\mathbf{I}_1$ as an example, we produce the multiscale feature embeddings as:

$$\{\varphi_1^{s=2}, \varphi_1^{s=4}, \varphi_1^{s=8}\} = E_\theta(\mathbf{I}_1). \tag{1}$$

For the input image $\mathbf{I}_1$ of resolution $H \times W$, the scale s indicates a feature map of resolution $H/s \times W/s$.

Cross-Frame Global Matching The cross-frame global matching module (CFGM) is designed to accomplish coarse-scale geometric matching. To benefit from the MIM pretext task, we first process the scale $s = 8$ feature map $\varphi_1^{s=8}$ with the transformer block [27]:

$$\{\overline{\varphi}_1^{s=8'}, \overline{\varphi}_2^{s=8'}\} = T_\theta(\varphi_1^{s=8}, \varphi_2^{s=8}). \tag{2}$$

In the pretraining stage, the masked feature map is recovered by the appended transformer blocks. Then, we follow

LoFTR [48] in using linear transformer blocks to correlate the source and support frame feature:

$$\{\overline{\varphi}_1^{s=8}, \overline{\varphi}_2^{s=8}\} = L_\theta(\varphi_1^{s=8'}, \varphi_2^{s=8'}). \tag{3}$$

To compute the global matching results, we first compute the 4D correlation volume $\mathbf{C}\left(\overline{\varphi}_1^{s=8}, \overline{\varphi}_2^{s=8}\right) \in \mathbb{R}^{H/8\times W/8\times H/8\times W/8}$, where:

$$C_{ijkl} = \sum_h \frac{1}{\gamma}\left(\overline{\varphi}_1^{s=8}\right)_{ijh} \cdot \left(\overline{\varphi}_2^{s=8}\right)_{klh}, \tag{4}$$

where γ is a temperature scalar. The coarse matches are computed as a summation over pixel locations $\mathbf{X} \in \mathbb{R}^{(H/8)(W/8)\times 2}$ weighted by the softmaxed correlation volume. That is, after the correlation volume $\mathbf{C}$ being reshaped to $\mathbf{C} \in \mathbb{R}^{(H/8)(W/8)\times(H/8)(W/8)}$, we apply the softmax:

$$\widetilde{C_{ij}} = \text{softmax}(C_{ij}). \tag{5}$$

Here, element C_{ij} is a size $(H/8)(W/8) \times 1$ vector. We conclude the coarse global matching results as:

$$T_*^{s=8} = \widetilde{\mathbf{C}} \times \mathbf{X}. \tag{6}$$

Note, Eqn. 6 will cause ambiguous results when multiple similar textureless local patches exist, *i.e.*, multiple peak values in softmaxed correlation vector $\widetilde{C_{ij}}$. To resolve this, we modify Eqn. 6 with:

$$T_*^{s=8}, P_*^{s=8} = D_\theta\left(\widetilde{\mathbf{C}} \times M(\mathbf{X})\right), \tag{7}$$

where $M(\mathbf{X})$ is cosine positional embeddings with learnable tokens [20, 48], projecting the 2D pixel locations to a high dimensional space to avoid ambiguity when multiple similar patches exist. The decoder D_θ decodes $T_*^{s=8}$, initial correspondences estimation at scale $s = 8$, and $P_*^{s=8}$, initial confidence estimation.

Multi-Scale Refinement We follow [20] in using the multi-scale refinement module:

$$\Delta T^s, \Delta P^s = R_\theta(\varphi_1^s, f(\varphi_2^s, T^s)), \tag{8}$$

where function $f(\cdot)$ indicates the bilinear interpolation to align the support frame feature using the current estimated correspondences T^s, shown in Fig. 2. To accommodate the transfer between pretexting and finetuning stage, we apply depth-wise convolution [20] in R_θ. We detail the discussion in Fig. 3 and Sec.3.2. The correspondences and confidence on the next scale are initialized with the bilinear upsampling.

3.2. Paired MIM Pretraining

Paired Masked Image Modeling (MIM) MIM is extensively adopted in image classification task [23, 63]. An image classification network can be further improved after MIM pretexting. As shown in Fig. 1 and 4, the network reconstructs the input from randomly masked feature embeddings at a specific scale. In this work, we investigate the benefit of pretraining both the encoder and decoder under MIM. Compared to only pretraining the encoder, pretraining the whole network further reduces the domain gap between pretexting and finetuning tasks.

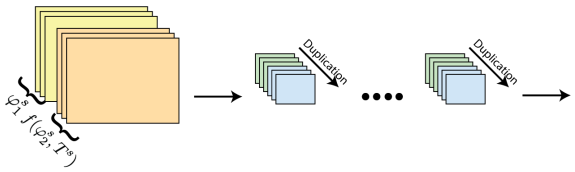


Figure 3. **Resolution of the Discrepancy between R_θ and R'_θ.** We adopt stacks of the depth-wise convolution in the refinement module, *i.e.*, each convolution kernel only works with one channel of the input feature maps. This makes refiner R'_θ in pretexting a sub-graph of refiner R_θ in finetuning. While transferring from the pretexting task to finetuning task, the input feature map concatenates an extra aligned support frame feature $f(\varphi_2^s, T^s)$. As the bilinear sampling f imposes minimal distribution change, we duplicate the kernel weight along the channel dimension.

Masking Strategy We follow SimMIM [63] in using randomly selected 32×32 mask patches with a predefined masking ratio r_1 and r_2 for source and support frames. For source view, given the feature embeddings $\varphi_1^{s=2}$ output by the extractor E_θ at scale $s = 2$, we apply the randomly generated mask $\mathbf{w}$ to mask out the feature embeddings, *i.e.*:

$$\varphi_1^{s=2'} = \varphi_1^{s=2} * (1 - \mathbf{w}) + \mathbf{x} * \mathbf{w}, \tag{9}$$

where $\mathbf{x}$ is the learnable mask tokens. Note, our extractor E_θ starts from a 3×3 convolution kernel to avoid leakage of the masked patches.

Prediction Heads Different from SimMIM [63], our prediction heads include most network components of the decoder. We complete the masked feature embeddings with the transformer as:

$$\varphi_1^{s=8'} = T_\theta(\varphi_1^{s=8}). \tag{10}$$

Here, we use the same notation as Eqn. 2 since both indicate image features at the scale $s = 8$. Note that the subsequent network component LoFTR is a series of linear transformer blocks [27] which reduce the quadratic computational complexity to linear. However, empirically we find the linear transformer poorly recovers the masked patches. We thus append the transformer blocks.

As shown in Fig. 2, after Eqn. 10, we feed the completed feature map to CFGM. Note the refiner between the two stages is different. Instead of taking a stacked feature map (Eqn. 8), in pretexting we only take in a single feature map:

$$\Delta\mathbf{I}_1^s = R'_\theta(\varphi_1^s), \quad \Delta\mathbf{I}_2^s = R'_\theta(\varphi_2^s). \tag{11}$$

To account for the difference between Eqn. 8 and Eqn. 11, we apply depth-wise convolution, where each convolution kernel operates on one channel of the feature map, shown in Fig. 3. Since $f(\varphi_2^s, T^s)$ in Eqn. 8 is a resampled support frame feature, it imposes minimal distribution difference to φ_2^s. Then, while transferring from the pretexting task to the downstream task, we only need to duplicate the channel of R_θ to complete the initialization. We follow SimMIM [63] in estimating full resolution residual RGB images in each scale of the decoder. We visualize the reconstructed paired masked images in Fig. 4.

Network Components not included in pMIM Since the feature map at $s = 2$ contains little information about masked patches, the pretraining only includes refinement modules at scale $s = 4$ and $s = 8$. Furthermore, the CFGM decoder D_θ and part of R_θ are not included. We pretrain the rest network component with synthetic image pairs [54].

Prediction Objective Set the accumulated reconstruction at each scale s as $\mathbf{I}^s$, we regress the raw pixel value with an l_1 loss:

$$\mathcal{L}_M = \sum_s \frac{1}{N}(|\mathbf{I}_1^s - \mathbf{I}_1|_1 + |\mathbf{I}_2^s - \mathbf{I}_2|_1), \tag{12}$$

where N is the number of unmasked pixels.

3.3. Dense Geometric Matching Loss

Homography Loss The image correspondences between two planar structures are constrained by a 3×3 homography matrix $\mathbf{H}$ with 8 DoF. Compared to correspondences estimation over arbitrary shapes, the correspondences in planar structures possess a lower rank. Given a surface normal $\mathbf{n}$ computed using the depth gradient [36], the homography of the pixel can be computed as:

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}_1^\intercal \\ \mathbf{h}_2^\intercal \\ \mathbf{h}_3^\intercal \end{bmatrix} = \mathbf{K}_1 \left(\mathbf{R} + \frac{\mathbf{t}^\top}{d} \mathbf{n} \right) \mathbf{K}_2^{-1}, \tag{13}$$

where the $\mathbf{K}_1$ and $\mathbf{K}_2$ are intrinsic matrices of $\mathbf{I}_1$ and $\mathbf{I}_2$, $\mathbf{R}$ and $\mathbf{t}$ are camera rotation and translation, and d is the pixel depth. We randomly sample K anchor points $\{\mathbf{p}_m \mid 1 \le m \le K\}$. For each anchor point $\mathbf{p}_m$, we sample K candidate points $\{\mathbf{q}_n^m \mid 1 \le n \le K\}$. We determine a co-planar indicator matrix $\mathcal{O}^+$ of size $K \times K$ to suggest all co-planar pairs. We use the normal consistency, point-to-plane distance, and homography consistency to compute the co-planar groundtruth, detailed in Supp. Finally, we apply a gradient-based penalty, penalizing the correspondences difference between the estimation and the groundtruth.

$$\mathcal{L}_h^s = \frac{1}{|\mathcal{O}^+|} \sum_{\mathcal{O}^+_{\mathbf{p},\mathbf{q}}=1} |\left(T_{\mathbf{p}}^s - T_{\mathbf{q}}^s\right) - \left(\overline{T}_{\mathbf{p}}^s - \overline{T}_{\mathbf{q}}^s\right)|_1. \tag{14}$$

Global Matching Loss Following [48], we minimize a binary cross-entropy loss over the correlation volume $\mathbf{C}$ after a dual-softmax operation:

$$\widetilde{C_{ijkl}}' = \text{softmax}(C_{ij}) \cdot \text{softmax}(C_{kl}), \tag{15}$$

where C_{ij} and C_{kl} are $(H/8)(W/8) \times 1$ vectors. The loss is defined as:

$$\begin{aligned} \mathcal{L}_g = &- \frac{1}{|\mathcal{M}^+|} \sum_{ijkl \in \mathcal{M}^+} \log \widetilde{C_{ijkl}}' \\ &- \frac{1}{|\mathcal{M}^-|} \sum_{ijkl \in \mathcal{M}^-} \log \left(1 - \widetilde{C_{ijkl}}'\right), \end{aligned} \tag{16}$$

where $\mathcal{M}^+$ and $\mathcal{M}^-$ are groundtruth indicator matrix of size $H \times W \times H \times W$ indicating whether a source frame pixel (i, j) pairs with a target frame pixel (k, l).

Refinement Loss Following [20], we supervise both correspondences and confidence on each scale of the predictions,

$$\mathcal{L}_r^s = \frac{1}{|P^+|} \sum_{ij \in P^+} \left| T_{ij}^s - \overline{T}_{ij}^s \right|_2, \tag{17}$$

where P_{ij}^+ is a $H \times W$ matrix that indicates whether a valid pair is found at pixel location ij in the source frame. Similarly, the loss of confidence is defined as:

$$\mathcal{L}_c^s = -\frac{1}{|\mathcal{P}^+|} \sum_{ij \in P^+} \log(P_{ij}) - \frac{1}{|\mathcal{P}^-|} \sum_{ij \in P^-} \log(1 - P_{ij}). \tag{18}$$

Total Loss The total loss is a weighted summation of proposed losses:

$$\mathcal{L} = \frac{1}{4} \sum_s (L_r^s + w_c \mathcal{L}_c^s) + w_g \cdot \mathcal{L}_g + \frac{1}{4} w_h \sum_s \mathcal{L}_h^s. \tag{19}$$

The constant 4 comes from the four scales $s = \{1, 2, 4, 8\}$ set in our paper.

4. Experiments

We first compare with other SoTA dense matching methods on the MegaDepth dataset. Then, to comprehensively reflect the contributions from both the density and accuracy of geometric matching, we follow [20, 48] in using the two-view relative camera pose estimation performance as the metric. We report on both the outdoor scenario MegaDepth [30] dataset and the indoor scenario ScanNet [12] dataset. We additionally evaluate on the HPatches [1] and the YFCC100m [51] datasets to demonstrate the generalizability of the model.

4.1. Implementation Details

Pretext stage From DeMoN [57], BlendedMVS [65], HyperSim [41], ARKitScenes [4], and TartanAir [60] datasets,

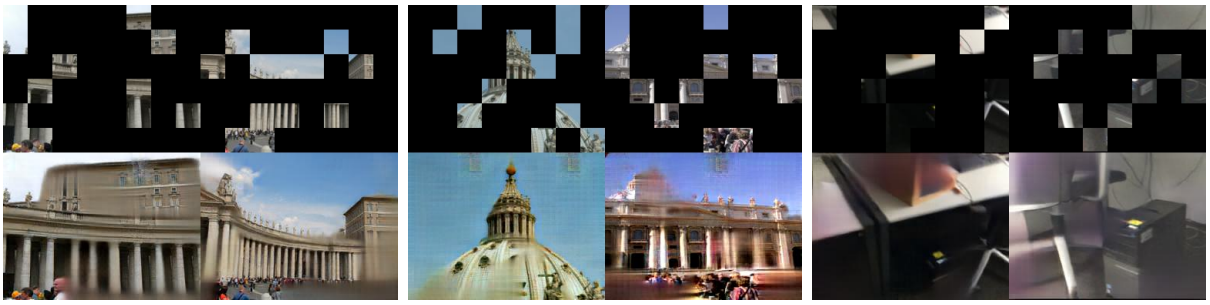


Figure 4. **Visual Quality of the paired MIM pretext task.** Visualized cases are from the MegaDepth and the ScanNet dataset.

we collect a pretraining dataset of $1,281,167$ image pairs, *i.e.*, the same size as ImageNet [14]. Each pair is collected with a fixed frame index interval. In the pretraining dataset, we train the model using a batchsize of 128 under the resolution 192×256. We use the Adam optimizer [28] with a learning rate $2e^{-4}$, running for 250k steps on $2\times$ A100 GPUs. We stack 1 transformer layer. We initialize the masking ratio $r_1 = 75\%$ and $r_2 = 75\%$. The masking operation applies to the ResNet, causing significantly different batch statistics between masked and unmasked inputs. Since the downstream task takes the unmasked image, we linearly reduce the support frame masking ratio r_2 to 0 and use a different batch normalization layer for support view, resolving the batch statistics difference. We also apply the synthetic image pair augmentation introduced in [54].

Finetuning stage Our model trains with a batchsize of 16 at the resolution 544×720. The learning rate is set to $4e^{-4}$, running 250k steps with a warmup of 25k steps. On $4\times$ A100 GPUs, we train for 5 days with the Adam optimizer. We follow [48] in sampling the paired images, weighted by the sequence length and overlap ratio. The softmax temperature γ is 0.1. We set loss weight w_g to 0.7 and w_h to 0.02. We sample 600×600 points for homography loss L_h.

4.2. Datasets

MegaDepth MegaDepth [30] collects over 10 thousand images of worldwide landmarks from the Internet. The collected images are processed by COLMAP [45] to produce groundtruth poses and depthmaps. The dataset collects images of significant visual contrast due to lighting conditions, view angles, and imaging devices. This imposes challenges to geometric matching.

ScanNet [12] is a large-scale indoor dataset with $1,613$ videos captured by RGB-D cameras. There are challenging textureless indoor scenes for geometric matching.

YFCC100m [51] is a large multi-media dataset. A subset of 72 reconstructions of tourist landmarks is generated with groundtruth poses and depthmap.

Hpatches [2] provides the pair of one source and five support images taken under different view angles and lighting conditions with groundtruth homography transformation.

Methods	Venue	Dense Match PCK ↑			Run-time (ms)
		@1 px	@3 px	@5 px	
RANSAC-FLow [46]	ECCV'20	53.47	83.45	86.81	3,596
PDC-Net [67]	CVPR'21	71.81	89.36	91.18	1,017
PDC-Net+ [54]	Arxiv'21	**74.51**	**90.69**	**92.10**	1,017
LIFE [26]	Arxiv'21	39.98	76.14	83.14	**78**
GLU-Net-GOCor [52]	NeurIPS'20	57.77	78.61	82.24	**71**
PDC-Net [67]	CVPR'21	68.95	84.07	85.72	88
PDC-Net+ [54]	Arxiv'21	72.41	86.70	88.12	88
PMatch (Ours)	CVPR'23	**79.83**	**95.18**	**96.52**	124

Table 1. **MegaDepth Dense Geometric Matching.** The running time of all methods is measured at the resolution 480×480. The upper and lower groups are methods running multiple or single times. [Key: **Best**, **Second Best**]

Category	Methods	Venue	Pose Estimation AUC ↑		
			@5°	@10°	@20°
Sparse W/ Detector	SuperGlue [44]	CVPR'19	42.2	61.2	75.9
	SGMNet [29]	Pattern'20	40.5	59.0	72.6
Sparse Wo/ Detector	DRC-Net [32]	ICASSP'22	27.0	42.9	58.3
	LoFTR [48]	CVPR'21	52.8	69.2	81.2
	QuadTree [49]	ICLR'22	54.6	70.5	82.2
	MatchFormer [59]	ACCV'22	53.3	69.7	81.8
	ASpanFormer [9]	ECCV'22	55.3	71.5	83.1
Dense	PDC-Net+ [54]	Arxiv'19	43.1	61.9	76.1
	DKM [20]	CVPR'23	**60.5**	**74.9**	**85.1**
	PMatch (Ours)	CVPR'23	**61.4**	**75.7**	**85.7**

Table 2. **MegaDepth Two-View Camera Pose Estimation.** We compare three groups of methods following SuperGlue [44] in evaluation. The pose AUC error is reported. Our method shows substantial improvement. [Key: **Best**, **Second Best**]

4.3. Dense Geometric Matching

We follow the RANSAC-Flow [46] in training and testing split on the MegaDepth dataset. The PCK scores in Tab. 1 refer to the thresholded keypoints accuracy. We divide the baseline methods into single and multiple run methods. Note, the baseline methods PDC Net [55] and PDC Net+ [54] consume the additional synthetic data generated using COCO [31] instance segmentation label. For PCK @1px, we outperform the SoTA single and multiple run methods by an absolute margin of 4.89% and 6.99% respectively. Meanwhile, we are about **8**$\times$ **faster** than SoTA

Figure 5. **Visual Quality of the Reconstruction.** We visualize 4 reconstructed images using estimated dense correspondences. In each group, from left to right is the source image, support image, and the reconstructed image. The areas of low confidence are filled with white color. In ScanNet where the confidence groundtruth is not available, we use forward-backward flow consistency mask as a replacement.

baselines while suppassing SoTA performance.

4.4. Two-View Camera Pose Estimation

Evaluation Protocol In the MegaDepth, ScanNet, and Hpatches datasets, we follow the evaluation protocol of [20, 44, 48] in reporting the pose accuracy AUC curve thresholded at 5, 10, and 20 degrees. In the YFCC100m dataset, we follow the protocol of RANSAC-Flow [46], additionally reporting the pose mAP value. The pose estimation is considered an outlier if its maximum degree error of translation or rotation exceeds the threshold. The two-view relative pose is estimated using the five-point algorithm [37] with RANSAC [15] via the OpenCV implementation [6].

Baseline Methods We compare with three groups of the methods, *i.e.*, sparse methods with detector [29, 44], sparse methods without detector [9, 32, 48, 49, 59] and dense methods [13, 20, 46, 54, 55, 61]. For sparse detector based methods, we use SuperPoint [16] as the keypoint detector. For dense methods, we further categorize them into single-run and multiple-run methods. For multiple-run methods, *e.g.*, RANSAC-Flow [46], it repeats the prediction while reducing the visual difference with an estimated homography transformation. Among baselines, AspanFormer [9] is a recent publicly available sparse detector-free method, improving LofTR with a sophisticated attention mechanism.

Outdoor Dataset We test our method on the outdoor dataset MegaDepth. We follow the training and validation split of [20, 44, 48]. The evaluation split contains $1,500$ paired images randomly selected from the scene 0015 and 0022. As shown in Tab. 2, we achieve an absolute improvement of 0.9% over the recent SoTA dense method DKM [20]. Compared to the SoTA sparse method ASpanFormer [9], we maintain an improvement of 6.1%.

Indoor Dataset We test our method on the indoor dataset ScanNet. We follow [20] in training and testing protocol, resizing images to 480×640. The validation split of ScanNet consists of $1,500$ image pairs [44]. In Tab. 3, we maintain competitive performance with the SoTA dense method DKM [20] and outperform SoTA sparse method by 1.4%.

Category	Methods	Venue	Pose Estimation AUC ↑		
			@5°	@10°	@20°
Sparse W/ Detector	SuperGlue [44]	CVPR'19	16.2	33.8	51.8
	SGMNet [29]	PR'20	15.4	32.1	48.3
Sparse Wo/ Detector	DRC-Net [32]	ICASSP'22	7.7	17.9	30.5
	LoFTR [48]	CVPR'21	22.0	40.8	57.6
	QuadTree [49]	ICLR'22	24.9	44.7	61.8
	MatchFormer [59]	ACCV'22	24.3	43.9	61.4
	ASpanFormer [9]	ECCV'22	**25.6**	46.0	63.3
Dense	PDC-Net+ [54]	Arxiv'19	20.2	39.4	57.1
	DKM [20]	CVPR'23	**29.4**	**50.7**	**68.3**
	PMatch (Ours)	CVPR'23	**29.4**	**50.1**	**67.4**

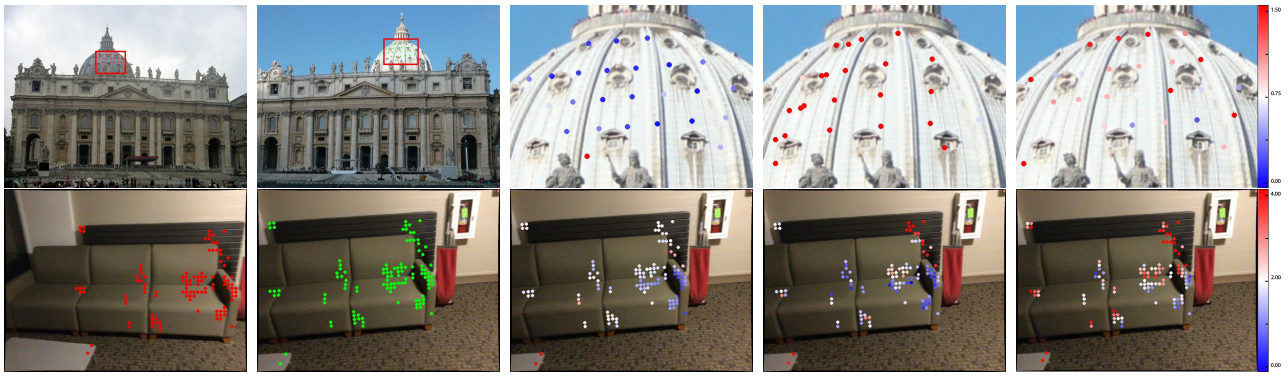
Table 3. **ScanNet Two-View Camera Pose Estimation.** We follow SuperGlue [44] in the testing protocol. The pose AUC error is reported. Our method achieves clear improvement over other baselines. [Key: **Best**, **Second Best**]

Methods	Venue	Pose Estimation AUC ↑			Pose Estimation mAP ↑		
		@5°	@10°	@20°	@5°	@10°	@20°
RANSAC-Flow [46]	ECCV'20	-	-	-	64.9	73.3	81.6
PDC-Net [55]	CVPR'21	35.7	55.8	72.3	63.9	73.0	81.2
PDC-Net+ [54]	Arxiv'21	37.5	58.1	74.5	**67.4**	**76.6**	**84.6**
OANet [13]	ICCV'19	-	-	-	52.2	-	-
CoAM [61]	CVPR'21	-	-	-	55.6	66.8	-
PDC-Net [55]	CVPR'21	32.2	52.6	70.1	60.5	70.9	80.3
PDC-Net+ [54]	Arxiv'21	34.8	55.4	72.6	63.9	73.8	82.7
ASpanFormer [9]	ECCV'22	**44.5**	**63.8**	**78.4**	-	-	-
PMatch (Ours)	CVPR'23	**45.7**	**65.2**	**79.8**	**75.9**	**83.1**	**89.3**

Table 4. **YFCC100m Two-View Camera Pose Estimation.** The upper group runs multiple times, while the lower group runs a single time. We follow [67] in the evaluation and preprocessing, reporting both pose AUC and mAP errors. [Key: **Best**, **Second Best**]

Generalization to YFCC100m We use the MegaDepth trained model to test on YFCC100m [51] dataset. We follow the preprocessing steps of [67], evaluated on 4 scenes with a total of $1,000$ images. During the evaluation, we resample the input images of the shorter side to 480. Tab. 4 shows that our method can achieve a superior generalization ability, maintaining an improvement of 1.2% over SoTA sparse methods [9].

Generalization to HPatches Following LoFTR [48], we test the MegaDepth dataset trained model on HPatches.



(a) Source Frame $\mathbf{I}_1$ (b) Support Frame $\mathbf{I}_2$ (c) PMatch (Ours) (d) DKM [20] (e) LoFTR [48]

Figure 6. **Visual Comparisons.** We conduct the visual comparison against the SoTA dense [20] and sparse [48] methods on the MegaDepth and the ScanNet datasets. The color from blue to red indicates an increment in the end-point-error (L2 error).

Category	Methods	Venue	Pose Estimation AUC ↑		
			@3px	@5px	@10px
Sparse W/ Detector	D2Net [19]	CVPR'19	23.2	35.9	53.6
	R2D2 [40]	NeurIPS'19	50.6	63.9	76.8
	DISK [56]	NeurIPS'20	52.3	64.9	78.9
	SuperGlue	CVPR'19	53.9	68.3	81.7
Sparse Wo/ Detector	NCNet [42]	ECCV'20	48.9	54.2	67.1
	DRC-Net [32]	ICASSP'22	50.6	56.2	68.3
	LoFTR [48]	CVPR'21	65.9	75.6	**84.6**
Dense	DKM [20]	CVPR'23	**71.3**	**80.6**	**88.5**
	PMatch (Ours)	CVPR'23	**71.9**	**80.7**	**88.5**

Table 5. **Hpatches Homography Estimation.** We follow [48] in evaluation protocol. We report the corner point AUC error under the estimated homography matrix. [Key: **Best**, **Second Best**]

In evaluation, the homography matrix is estimated using OpenCV's implementation. We compare correspondences accuracy computed using the groundtruth and estimated homography. The image pairs in HPatches have lighting differences or view differences. The pattern is different from the training dataset MegaDepth. Under the unseen testing scenario, our model generalizes best among baselines.

5. Ablation Study

Qualitative Comparison The visual quality of reconstructed images using the predicted correspondences is visualized in Fig. 5. We conduct a visual comparison with other SoTA dense and sparse methods in Fig. 6. In Row 1, (c), and (d), compared to DKM [20], the proposed CFGM module achieves correct initial correspondences. In Row 1, (c), and (e), compared to LoFTR [48], multi-scale dense refinement improves fine-scale correspondence accuracy. In Row 2, (c), (d), and (e), our CFGM and homography loss achieve accurate correspondence estimation on textureless planar surface, *e.g.*, the black wall behind the sofa.

Running Time Evaluated on an RTX 2080 Ti GPU, we run 160 ms for an image of 480×640 while LoFTR [48] runs 116 ms and DKM [20] runs 148 ms. Our model runs similarly compared to the baselines. The running time comparison to other dense methods is in Tab. 1.

Baseline	CFGM	L_H	pMIM Encoder $(E_\theta, T_\theta, L_\theta)$	pMIM Decoder (R_θ)	Pose Estimation AUC ↑		
					@5°	@10°	@20°
✓					56.1	71.5	83.0
✓	✓				57.5	72.6	83.9
✓	✓	✓			57.9	72.9	84.1
✓	✓	✓	✓		60.6	75.0	85.3
✓	✓	✓	✓	✓	**61.4**	**75.7**	**85.7**

Table 6. **Ablation Studies on MegaDepth.** The baseline method is the network in Fig. 2 with only a LoFTR module, *i.e.*, without the other components of CFGM. The ablation is conducted under the same training and testing resolution as Tab. 2. Bold marks best.

Benefit of the paired MIM pretraining Shown in Tab. 6, with the paired MIM pretext task, the pose accuracy thresholded at 5° improves by $3.5\% = 61.4\% - 57.9\%$. A visual result of the paired MIM task is shown in Fig. 4.

CFGM and Homography Loss The benefit of the proposed CFGM module and homography loss L_h is included in Tab. 6. They help the network predict more accurate results in textureless planar surfaces.

6. Conclusion

This work investigates the benefit of pretraining the encoder and decoder of a dense geometric matching network under the paired MIM task. We solve the discrepancy between the pretraining and finetuning tasks. Also, we contribute an improved geometric matching network by reducing the ambiguity of textureless patches and augmenting the learning of local planar surfaces.

Limitation Our method does not produce robust local descriptors. When registering a keypoint, our method needs to run dense matching over all past frames, imposing latency for time-sensitive applications, *e.g.*, odometry estimation.

References

[1] Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *CVPR*, 2017. 5

[2] Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *CVPR*, 2017. 6

[3] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. In *ICLR*, 2022. 2

[4] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, et al. Arkitscenes–a diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data. *arXiv preprint arXiv:2111.08897*, 2021. 5

[5] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *CVIU*, 2008. 2

[6] Gary Bradski and Adrian Kaehler. Opencv. *Dr. Dobb's journal of software tools*, 2000. 7

[7] Samarth Brahmbhatt, Jinwei Gu, Kihwan Kim, James Hays, and Jan Kautz. Geometry-aware learning of maps for camera localization. In *CVPR*, 2018. 1

[8] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, pages 9650–9660, 2021. 2

[9] Hongkai Chen, Zixin Luo, Lei Zhou, Yurun Tian, Mingmin Zhen, Tian Fang, David Mckinnon, Yanghai Tsin, and Long Quan. Aspanformer: Detector-free image matching with adaptive span transformer. In *ECCV*, 2022. 1, 2, 6, 7

[10] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *ICML*, 2020. 2

[11] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 2

[12] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. 5, 6

[13] Luanyuan Dai, Xin Liu, Jingtao Wang, Changcai Yang, and Riqing Chen. Learning two-view correspondences and geometry via local neighborhood correlation. *Entropy*, 2021. 7

[14] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009. 2, 6

[15] Konstantinos G Derpanis. Overview of the ransac algorithm. *Image Rochester NY*, 2010. 7

[16] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPRW*, 2018. 1, 2, 7

[17] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 2

[18] Elan Dubrofsky. Homography estimation. *Diplomová práce. Vancouver: Univerzita Britské Kolumbie*, 2009. 1

[19] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-net: A trainable cnn for joint detection and description of local features. In *CVPR*, 2019. 1, 2, 8

[20] Johan Edstedt, Ioannis Athanasiadis, Mårten Wadenbäck, and Michael Felsberg. DKM: Dense kernelized feature matching for geometry estimation. In *CVPR*, 2023. 1, 3, 4, 5, 6, 7, 8

[21] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *PAMI*, 2017. 1

[22] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. In *NeuriPS*, 2020. 2

[23] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022. 2, 4

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2, 3

[25] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, 2017. 2

[26] Zhaoyang Huang, Xiaokun Pan, Runsen Xu, Yan Xu, Guofeng Zhang, Hongsheng Li, et al. Life: Lighting invariant flow estimation. *arXiv preprint arXiv:2104.03097*, 2021. 6

[27] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *ICML*, 2020. 3, 4

[28] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6

[29] Jianan Li, Xuemei Xie, Qingzhe Pan, Yuhan Cao, Zhifu Zhao, and Guangming Shi. Sgm-net: Skeleton-guided multimodal network for action recognition. *PR*, 2020. 6, 7

[30] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *CVPR*, 2018. 5, 6

[31] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 6

[32] Jinjiang Liu and Xueliang Zhang. Drc-net: Densely connected recurrent convolutional neural network for speech dereverberation. In *ICASSP*, 2022. 1, 6, 7, 8

[33] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 1, 2

[34] Iaroslav Melekhov, Aleksei Tiulpin, Torsten Sattler, Marc Pollefeys, Esa Rahtu, and Juho Kannala. Dgc-net: Dense geometric correspondence network. In *WACV*, 2019. 2

[35] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *CVPR*, 2020. 2

[36] Yosuke Nakagawa, Hideaki Uchiyama, Hajime Nagahara, and Rin-Ichiro Taniguchi. Estimating surface normals with depth image gradients for fast and accurate registration. In *3DV*, 2015. 5

[37] David Nistér. An efficient solution to the five-point relative pose problem. *PAMI*, 2004. 7

[38] Yuki Ono, Eduard Trulls, Pascal Fua, and Kwang Moo Yi. Lf-net: Learning local features from images. In *NeurIPS*, 2018. 2

[39] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *CoRR*, 2018. 2

[40] Jerome Revaud, Cesar De Souza, Martin Humenberger, and Philippe Weinzaepfel. R2d2: Reliable and repeatable detector and descriptor. In *NeuriPS*, 2019. 1, 2, 8

[41] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *ICCV*, 2021. 5

[42] Ignacio Rocco, Relja Arandjelović, and Josef Sivic. Efficient neighbourhood consensus networks via submanifold sparse convolutions. In *ECCV*, 2020. 1, 2, 8

[43] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *ICCV*, 2011. 2

[44] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 2, 6, 7

[45] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 1, 6

[46] Xi Shen, François Darmon, Alexei A Efros, and Mathieu Aubry. Ransac-flow: generic two-stage image alignment. In *ECCV*, 2020. 3, 6, 7

[47] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 2

[48] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *CVPR*, 2021. 1, 2, 4, 5, 6, 7, 8

[49] Shitao Tang, Jiahui Zhang, Siyu Zhu, and Ping Tan. Quadtree attention for vision transformers. In *ICLR*, 2022. 6, 7

[50] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020. 1

[51] 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*, 2016. 5, 6, 7

[52] Prune Truong, Martin Danelljan, Luc V Gool, and Radu Timofte. Gocor: Bringing globally optimized correspondence volumes into your neural network. In *NeuriPs*, 2020. 3, 6

[53] Prune Truong, Martin Danelljan, and Radu Timofte. Glu-net: Global-local universal network for dense flow and correspondences. In *CVPR*, 2020. 3

[54] Prune Truong, Martin Danelljan, Radu Timofte, and Luc Van Gool. Pdc-net+: Enhanced probabilistic dense correspondence network. *arXiv preprint arXiv:2109.13912*, 2021. 1, 2, 3, 5, 6, 7

[55] Prune Truong, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning accurate dense correspondences and when to trust them. In *CVPR*, 2021. 1, 6, 7

[56] Michał Tyszkiewicz, Pascal Fua, and Eduard Trulls. Disk: Learning local features with policy gradient. In *NeuriPS*, 2020. 1, 2, 8

[57] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. Demon: Depth and motion network for learning monocular stereo. In *CVPR*, 2017. 5

[58] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *JMLR*, 2010. 2

[59] Qing Wang, Jiaming Zhang, Kailun Yang, Kunyu Peng, and Rainer Stiefelhagen. Matchformer: Interleaving attention in transformers for feature matching. In *ACCV*, 2022. 6, 7

[60] Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual slam. In *IROS*, 2020. 5

[61] Olivia Wiles, Sebastien Ehrhardt, and Andrew Zisserman. Co-attention for conditioned image matching. In *CVPR*, 2021. 7

[62] Tete Xiao, Mannat Singh, Eric Mintun, Trevor Darrell, Piotr Dollár, and Ross Girshick. Early convolutions help transformers see better. In *NeuriPs*, 2021. 2

[63] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *CVPR*, 2022. 2, 4, 5

[64] Ceyuan Yang, Zhirong Wu, Bolei Zhou, and Stephen Lin. Instance localization for self-supervised detection pretraining. In *CVPR*, 2021. 2

[65] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In *CVPR*, 2020. 5

[66] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. Lift: Learned invariant feature transform. In *ECCV*, 2016. 2

[67] 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 *ICCV*, 2019. 6, 7

[68] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. In *ICLR*, 2022. 2