PA-LoFTR: Local Feature Matching with 3D Position-Aware Transformer

Anonymous authors
Paper under double-blind review

Abstract

We propose a novel image feature matching method that utilizes 3D position information to augment feature representation with a deep neural network. The proposed method introduces 3D position embedding to a state-of-the-art feature matcher, LoFTR, and achieves more promising performance. Following the coarse-to-fine matching pipeline of LoFTR, we construct a Transformer-based neural network that generates dense pixel-wise matches. Instead of using 2D position embeddings for transformer, the proposed method generates 3D position embeddings that can precisely capture position correspondence of matches between images. Importantly, in order to guide neural network to learn 3D space information, we augment features with depth information generated by a depth predictor. In this way, our method, PA-LoFTR, can generate 3D position-aware local feature descriptors with Transformer. We experiment on indoor datasets, and results show that PA-LoFTR improves the performance of feature matching compared to state-of-the-art methods.

1 Introduction

Finding feature matching between images is an important task to many computer vision works, including camera calibration, structure from motion (SfM), visual localization, simultaneous localization and mapping (SLAM), stereo matching, etc. The accuracy of feature matching can greatly affect performance of tasks above. Generally, local feature matching problem can be solved with three stages: feature detection, feature description and feature matching. Most of the existing methods follow the pipeline in sequence to handle the feature matching problem. Feature detector narrows the focus to a set of interest points on images. Feature description phase generates corresponding descriptor for each interest point. Finally, correspondences between the interest points of images are found by feature matching algorithms. While current methods try to extract features based on visual information from images, they pay little attention to the 3D position information of features, which limits the performance of matching.

As the development of deep learning, a number of works have introduced deep architecture to image feature matching. Many methods still follow the classic pipeline, where some methods develop networks for feature detection and some focus on feature description or matching with learning process. Recently, several works have tried to develop detector-free deep architectures that can generate dense matches (Rocco et al. (2018); Li et al. (2020); Rocco et al. (2020)). Instead of the classic pipeline that requires a feature detector to pick sparse interest points from images, the detector-free methods can provide pixel-wise matches and select matches with high confidence. In this way, feature detection can be dropped as each pixel is regarded as a potential interest point to be matched. Several detector-free methods can have better performance for images with poor texture, complex patterns and illumination or large viewpoint change, where feature detector cannot determine enough interest points following the classic pipeline. Some methods, such as LoFTR (Sun et al., 2021) and CoTR (Jiang et al., 2021), introduce Transformer into image feature matching, that the network can have global receptive field compared with convolutional neural networks (CNNs). Taking advantage of the ability to relate two features anywhere from the image, the methods can achieve state-of-the-art performance. However, the detector-free methods that generate dense matches can still give false correspondences under challenging pose changes or repetitive patterns. Though the Transformer-based architectures can learn global relation between features, the current methods are still working

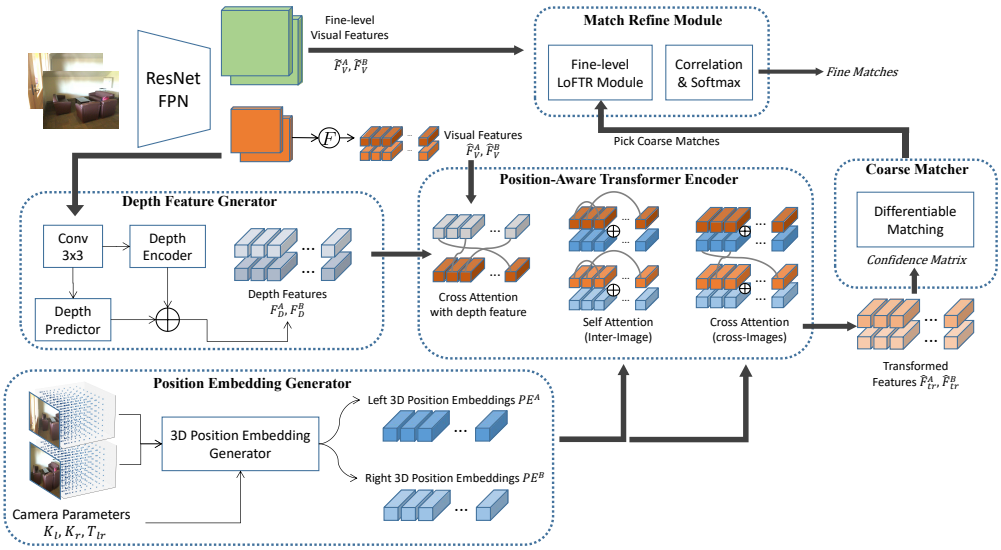


Figure 1: **Overview of Proposed method**. PA-LoFTR accepts a pair of images I_A, I_B, and uses a CNN backbone to give multi-level feature maps. The coarse-level feature maps F_V^A, F_V^B are augmented through three modules: **1.** Depth Feature Generator predicts a dense depth map for each feature map, and gives corresponding depth features F_D^A, F_D^B. **2.** Position Embedding Generator prepares 3D position embeddings PE^A, PE^B given camera intrinsic of each camera and extrinsic parameters between them. **3.** Position-Aware Transformer Encoder combines 3D position embeddings and depth features into visual features from backbone. The encoder layers contains one cross attention between F_V and F_D, one self attention for F_V^k with position embedding, and one cross attention for F_V^A and F_V^B with corresponding position embeddings.

with local visual features extracted directly from images, and most of them pay little attention to the 3D space information of feature points.

Based on the observations above, we propose 3D Position-Aware Local Feature Transformer (PA-LoFTR), a novel approach that encodes space information for local feature matching. We follow the coarse-to-fine matching idea of LoFTR to construct deep architecture. Inspired by the usage of position encoding for Transformer, we develop a 3D position embedding generator based on the coarse-to-fine structure of LoFTR. Instead of encoding 2D pixel coordinates to give position embeddings, we encode 3D point clouds for each pixel in images and transform them into the same world coordinates. When self and cross attention layers learn relations between local features from convolutional backbone, we add 3D position embeddings at each encoder layer in Transformer. Additionally, we construct a depth predictor branch to learn a depth distribution of a single image, and the depth feature learned will be passed in cross attention layer to co-relate with visual features from backbone. By encoding 3D position information and co-relating with depth features, PA-LoFTR can learn meaningful features containing both visual and 3D space information that greatly help determine concise matches between two images. We evaluate the proposed architecture with a indoor dataset on camera pose estimation and other image matching tasks. The experiments show that PA-LoFTR can provide high-quality matches under challenging scenarios, and achieve state-of-the-art performance on some tasks.

In this study, our main technique contributions are:

- We show that a 3D position encoder serving as position embedding generator for Transformer can greatly improve image correspondence quality determined by neural network.
- We propose a depth feature generator that give rough depth distributions for single image, which can help Transformer learn more space information and improve matching performance.

- We demonstrate that PA-LoFTR achieves state-of-the-art performance on indoor dataset and multiple tasks.

2 Related Works

Classic Feature Matching Pipeline. Classic feature matching pipeline includes feature detection, feature description and feature matching (Low (2004); Bian et al. (2017); Sattler et al. (2009); Tuytelaars & Gool (2000)). Many methods follow the classic feature matching pipeline to handle the image correspondence problem. SIFT (Lowe, 2004) introduces a way to build keypoint detector and descriptor. ORB (Rublee et al., 2011) improves the efficiency to produce descriptors. Given detected features and their descriptors, RANSAC can filter outliers to provide precise matches (Fischler & Bolles, 1987). Classic methods have proved their effectiveness and have been widely applied in the industry.

Feature Matching with Deep Learning Methods. As the deep learning methods spring up, the performance of feature matching has been significantly improved under challenging scenarios, such as large change of viewpoint and illumination. Several methods introduce deep architectures to improve one or more stages of classic feature matching pipeline. LIFT (Yi et al., 2016) constructs a CNNs-based network to build feature detector and feature descriptor. MagicPoint (Detone et al., 2017) makes extra efforts to predict the corners in images. They have been proved effective in detecting keypoints with meaningful descriptors. SuperPoint (DeTone et al., 2018) presents a self-supervised framework to learn interest point detectors and descriptors based on MagicPoint, and SuperPoint applies homographic adaptation to improve the performance of feature detection. Methods based on classic pipeline use metrics to optimize matches, such as nearest neighbor to search the matches. SuperGlue (Sarlin et al., 2020) introduces an attentional graph neural network to match the local features captured by feature detector and descriptor. Several methods accomplish feature matching tasks based on deep learning methods (Efe et al. (2021); Rocco et al. (2022); Zhou et al. (2021); Truong et al. (2020); Dusmanu et al. (2019)).

Recently, some researchers have introduced Transformer to image correspondence problem. COTR (Jiang et al., 2021) adopts the architecture of DETR (Carion et al., 2020) that adopts Transformer encoder and decoder, and it tackles image correspondence problem in the way of querying pixel positions. LoFTR (Sun et al., 2021) regards each pixel in the image as a potential keypoint and apply Transformer encoder to learn relative relations between features, and it builds a coarse-to-fine pipeline to give precise matches. STTR (Li et al., 2021) conducts experiments that show Transformer can also be applied to stereo matching. The Transformer-based methods for image matching have shown the advantage of the global receptive field of Transformer, which can learn relations between features anywhere from the images. LoFTR can produce high-quality matches in challenging image pairs with low-texture areas and repetitive patterns. However, the usage of 2D or 1D position embeddings in methods above has a drawback, where position embeddings can not represent space correspondences between images. In this study, we propose a method that improves the performance of LoFTR by introducing meaningful 3D position embedding.

Position Embedding in Transformer. Recently, Transformer-based methods have been introduced to tackle multiple computer vision tasks, including object detection (Carion et al. (2020); Liu et al. (2022); Li et al. (2022); Jain et al. (2022); Zhang et al. (2022)), semantic segmentation (Miao et al. (2022); Li et al. (2022); Qian et al. (2022)), image feature matching (Jiang et al. (2021); Hong et al. (2022); Mao et al. (2022)), etc. While CNNs have the ability to aggregate features in a local window, Transformer can have global receptive field with attention mechanism that builds relations between features anywhere from the input sequences.

However, Transformer regards the input sequence as a set of unordered features, which leads to missing important position related information. To make use of the position information, transformer introduces position embeddings to the input features, which significantly improve the performance of models. However, when it comes to image matching problem, the widely used 2D position embedding methods (Carion et al. (2020); Jiang et al. (2021)) may serve as drawbacks. There are several methods working on improving the effectiveness of position embedding. STTR (Li et al., 2021) proves that adding position embeddings to the features will introduce a position-depended term in attention calculation, which is unrelated to image content, and STTR applies relative position embedding to relieve the problem. PETR (Liu et al., 2022) introduces 3D position embeddings

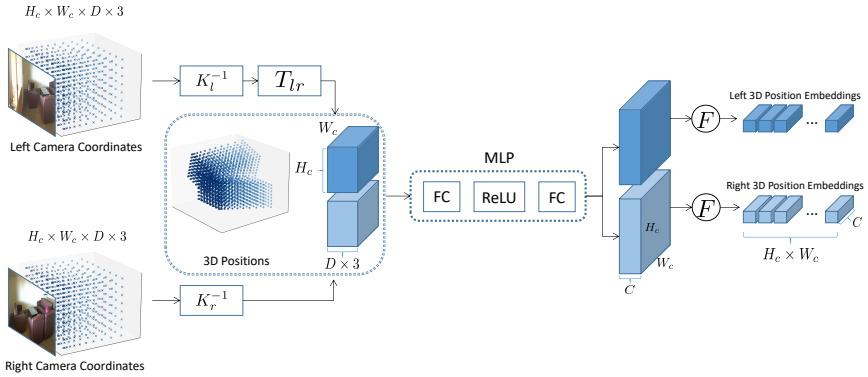


Figure 2: **Illustration of proposed 3D Position Embedding Generator.**For the pair of images given, a sparse 3D coordinate cloud is generated. Each coordinate represents a position at which a pixel can locate. Through camera intrinsic K_l, K_r and extrinsic T_{lr}, the coordinate clouds are aligned under the right camera coordinate. Then we use multi-layer perception to transform the coordinate clouds to 3D position embeddings. Before inputting to Transformer Encoder, a flatten operation Ⓕ is used to get position embedding sequences.

to enhance DETR3D (Wang et al., 2021). PETR transforms 3D world space to camera frustum space, and embeds 3D coordinates by a multi-layer perception to produce 3D position embeddings, which further generates 3D position-aware image features. Nevertheless, recent Transformer-based methods for image matching still pay little attention to constructing position embeddings related to 3D space. Our work introduces 3D position embeddings to Transformer encoder to generate 3D position-aware image features for better performance of matching.

3 METHODS

3.1 OVERVIEW OF ARCHITECTURE

Figure 1 gives the overall pipeline of proposed PA-LoFTR. Given a pair of input images I_A, I_B, the backbone network ResNet FPN (Lin et al., 2017a) extracts multi-level visual features. We mainly concern the feature maps at $1/8$ and $1/2$ of original image shape, and we follow LoFTR (Sun et al., 2021) to denote them as coarse-level visual features $\hat{F}_V^A, \hat{F}_V^B$ and fine-level visual features $\tilde{F}_V^A, \tilde{F}_V^B$. We follow the coarse-to-fine structure of LoFTR, where coarse-level features are mainly used to generate matches which are refined with fine-level features.

Our proposed method focuses on coarse-level matching and improves performance by augmenting $\hat{F}_V^A, \hat{F}_V^B$ with Position-Aware Transformer Encoder. Position-Aware Transformer accepts visual features $\hat{F}_V^A, \hat{F}_V^B$ from backbone, depth features F_D^A, F_D^B from depth feature generator, and 3D position embeddings as inputs, and outputs transformed features F_{tr}^A, F_{tr}^B. The transformed features are then used to calculate pair-wise similarity and keep coarse-level matches with high confidence. Finally, following LoFTR, the network looks into fine-level features around each pair of matches to refine final match positions.

3.2 3D POSITION EMBEDDING GENERATOR

Instead of using position embeddings from 2D pixel coordinates, we build the relation between 2D images and 3D space positions as shown in Figure 2. We first create meshgrid coordinates at the size of coarse-level feature maps, which is $H_c \times W_c$, by discretizing camera frustum space following DSGN (Chen et al., 2020) and PETR (Liu et al., 2022). We divide the depth space following linear-increasing discretization into D positions and get a meshgrid of size (H_c, W_c, D) where each cell represents a 3D position. The depth space is limited within $[d_{min}, d_{max}]$ along the center axis of

camera and D depth values are sampled, where each depth d_i can be calculated as

$$d_{i,i\in\{0,1...,D-1\}} = d_{min} + \delta_d \times i \times (i+1), \text{where } \delta_d = \frac{d_{max} - d_{min}}{D(D+1)}. \tag{1}$$

Each 3D coordinate in the meshgrid $\boldsymbol{P}$ with pixel (u, v) on the image can be represented as $p_i = (u \times d_i, v \times d_i, d_i)^T$. Further, as the image pair should share the same world space, the 3D coordinates are projected with camera intrinsic K_l, K_r and extrinsic T_{lr} where coordinate meshgrid of left image $\boldsymbol{P}_{left}$ is aligned with meshgrid of right image $\boldsymbol{P}_{right}$. The transformation to world space for a point follows Eq. 2, where camera extrinsic T_{lr} is divided into rotation R_{lr} and translation t_{lr}. Similar to PETR, we further normalize the 3D coordinates to a region of interest $[x_{min}, x_{max}, y_{min}, y_{max}, z_{min}, z_{max}]$, which depends on the dataset.

$$\begin{aligned} p_l^{3d} &= R_{lr}K_l^{-1}p_l + t_{lr}, \text{where } p_l \in \boldsymbol{P}_{left} \\ p_r^{3d} &= K_r^{-1}p_r, \text{where } p_r \in \boldsymbol{P}_{right} \end{aligned} \tag{2}$$

After transformation to aligned world coordinates, the point cloud matrix $\boldsymbol{P}^{3d} \in \mathbb{R}^{H_c \times W_c \times (D \times 3)}$ is got for each image. Given the point cloud as input, a multi-layer perception (MLP) network generates 3D position embedding (PE) which is flatten to a sequence $PE \in \mathbb{R}^{(H_c \times W_c) \times C}$, where C is channel dimension. The 3D PE of each image is then input to Position-Aware Transformer Encoder which is described in section 3.4.

3.3 Depth Feature Generator

The purpose of Depth Feature Generator is to provide additional global scene information during learning of Position-Aware Transformer Encoder. In 3D Position Embedding Generator, the network constructs the 3D position relation between pixel and world space, but it lacks explicit depth information which is vital to learning the global scene. We propose a depth prediction branch after the backbone network as shown in Figure 3. Given the coarse-level feature map $\hat{F_V}$ of a single image, Depth Feature Generator predicts a dense depth distribution and outputs depth features for each pixel of the feature map.

Coarse-level feature is first going through a convolutional layer. The output features then walk through two branches where a Depth Encoder extract depth features f_D with a single layer of Transformer encoder, and a Depth Predictor explicitly generates a dense depth distribution map. The depth encoder applies self-attention mechanism of Transformer, which will be described in detail in section 3.4, to learn the depth features. The dense depth map is used to generate depth embeddings similar to the way in MonoDETR (Zhang et al., 2022) and MonoDTR (Huang et al., 2022). To supervise learning of depth features, we follow the similar way in 3D Position Embedding Generator where depth space is discretized with LID into $k_{bins} + 1$ bins. The first k_{bins} represents the foreground depth values and the last one is for background. The category of a ground-truth depth label can be calculated as

$$k = -0.5 + \sqrt{0.25 + \frac{d - d_{min}}{\delta_d}}, \text{ where } \delta_d = \frac{d_{max} - d_{min}}{k_{bins}(k_{bins}+1)}. \tag{3}$$

We use a 1×1 convolutional layer to predict the depth category at each pixel and get a depth value through summation of results of softmax operation. We apply Focal loss (Lin et al., 2017b) to supervise category of depth for each pixel. Given the dense depth map d_{map}, we further interpolate the depth value among a learnable embedding pool to give depth embeddings P_D. The final depth features are generated as $F_D = f_D + P_D$ where $F_D \in \mathbb{R}^{(H_c \times W_c) \times C}$.

3.4 Position-Aware Transformer Encoder

After 3D position embedding generator and depth feature generator, Position-Aware Transformer Encoder aggregates all the information to give transformed features for matching. With position-aware feature embedding, the transformed features F_{tr}^A, F_{tr}^B are more easily to match.

Here we introduce the basic idea of Transformer (Vaswani et al., 2017) as background. A Transformer encoder can contain a sequence of Transformer encoder layers, and the core component of a Transformer is the attention layer. The attention layer accepts three types of vectors as input, which

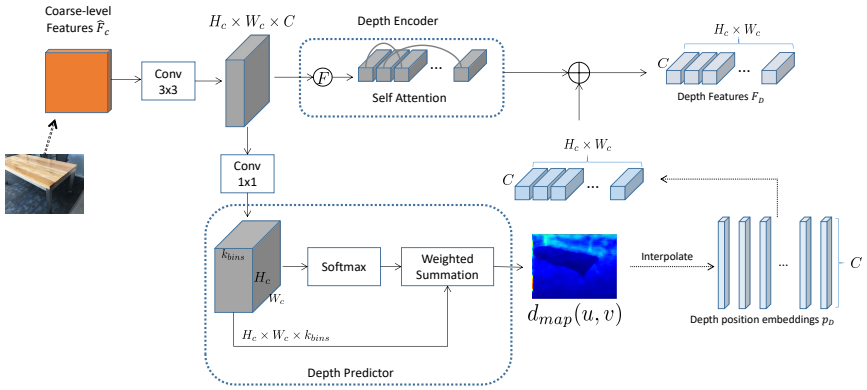


Figure 3: **Depth Feature Generator in PA-LoFTR**. Depth Feature Generator provides depth feature for each image of image pairs. Coarse-level visual feature $\hat{F}_V$ is first transformed by a convolutional layer, and go into two branches. **1.** Depth predictor gives a dense depth map d_{map} over the feature map, and generates depth embeddings p_D for each pixel. **2.** Depth Encoder apply single Transformer encoder layer and combine depth embeddings to generate depth features F_D.

are conventionally named as query, key, value. The key idea is that query vector Q will extract information from value vectors V based on the weights got from dot product with key vectors K where each value vector has its corresponding key vector. The dot product serves as a similarity score between query and keys which enables query Q to select more relevant values. The formal equation of attention layer computes as

$$Attn(Q, K, V) = \text{softmax}(QK^T)V. \tag{4}$$

Many studies have made efforts in designing effective and efficient attention layers including multi-scale deformable attention (Zhu et al., 2020), Quad-Tree Attention (Tang et al., 2022), etc. In our work, we follow LoFTR and apply Linear Attention (Katharopoulos et al., 2020) in the Transformer Encoder, which reduces the computation cost of Transformer to $O(N)$ with the length of input sequences.

In our proposed Position-Aware Transformer Encoder, each Transformer Encoder layer contains three stages as shown in Figure 1. The first cross attention layer accepts visual features $\hat{F}_V^i$ as queries and depth features F_D^i as keys and values, and here $i \in \{A, B\}$ means the first or second of image pair. The middle self attention layer builds relation for features across a single image, where it accepts $\hat{F}^i$ ($\hat{F}_A$ or $\hat{F}_B$ out of the first cross attention layer) as queries, keys and values. The last cross attention layer relates features between the pair of images, where $\hat{F}_A$ or $\hat{F}_B$ serves as queries and $\hat{F}_B$ or $\hat{F}_A$ as keys and values. Most importantly, position embeddings are vital to learning of Transformer Encoder. Different from LoFTR where 2D PE is used once at the beginning of Transformer Encoder, we apply 3D PE to the middle self attention and the last cross attention in each of the Transformer Encoder layers.

After feature Transformer Encoder, the transformed features $\hat{F}_{tr}^A, \hat{F}_{tr}^B$ are input to coarse-level matching module to select confident matches. A score matrix is calculated between two set of features where $S(i, j) = \frac{1}{\tau} \cdot \langle \hat{F}_{tr}^A(i), \hat{F}_{tr}^B(j) \rangle$, and we follow LoFTR to apply dual-softmax operator (Sun et al., 2021) to S to obtain mutual nearest neighbor matching by the matching probability $P_c(i, j) = \text{softmax}(S(i, \cdot))_j \cdot \text{softmax}(S(\cdot, j))_i$. With coarse matches selected, coarse-to-fine module applies spatial expectation to each pair of coarse matches to get final precise matches.

3.5 Implementation Details

Following LoFTR, we trained PA-LoFTR on the indoor dataset ScanNet (Dai et al., 2017). As mentioned in LoFTR, the original LoFTR model is trained with 64 GTX 1080Ti GPUs with full ScanNet dataset containing 1513 scenes, which requires too much resources and time cost for researchers. In this study, we apply a lite training setting, where we split partial of ScanNet which

Method	KITTI 2015
	EPE (px)
GA-Net (Zhang et al., 2019)	0.89
GWC-Net (Guo et al., 2019)	0.97
Bi3D (Badki et al., 2020)	1.16
PA-LoFTR (ours)	1.84
STTR (Vanilla Transformer) (Li et al., 2021)	0.45
STTR (with our 3D PE generator)	/

Table 1: Stereo Matching Results.

contains 100 scenes for training with 4 GTX 1080Ti GPUs and a batch size of 8, and test on the data split following Sarlin et al. (2020). To boost training, we use the pretrained CNN backbone from LoFTR, and the model parameters left in the following network are initialized with random weights, and the dimension of features is set as 128 instead of 256 to save memory. We further test model on KITTI Stereo dataset (Geiger et al., 2013), where we fine-tune the PA-LoFTR trained on ScanNet. As we notice that in many tasks, camera extrinsic is hard to get, which is a limit of our model, we implement a pose-refine version of PA-LoFTR with a simplified LoFTR branch to estimate camera extrinsic.

4 Experiments

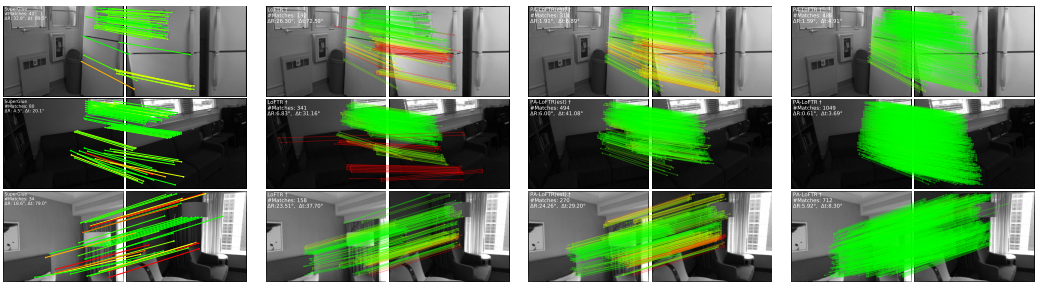
We experiment our method on two representative tasks, including stereo matching and relative pose estimation. Both tasks require the algorithm to extract strong matches to get precise disparity or relative pose. For pose estimation, we propose a pose-refine structure where camera extrinsic is first estimated and refined by PA-LoFTR to get the final pose. PA-LoFTR shows that it can get matches of high quality with the help of 3D position embedding generator.

4.1 Stereo Matching

We choose KITTI dataset to evaluate PA-LoFTR and the 3D Position Embedding Generator. We trained a PA-LoFTR model on ScanNet following our lite training setting, and we fine-tune the model on KITTI Stereo dataset. For fair comparison, we do not refine model on test dataset. Instead we fine-tune the model on KITTI object detection dataset with pseudo ground truth disparity following disp-RCNN (Sun et al., 2020). Though the model structure of PA-LoFTR is not designed for stereo matching where matches only occur on horizontal axis of images, we find PA-LoFTR can directly generate matches along horizontal lines with high quality. We calculate a sparse disparity with generated matches and evaluate on KITTI dataset. Compared to state-of-the-art methods targeting at disparity estimation, PA-LoFTR have limited performance due to its design of model architecture. To show the effectiveness of 3D Position Embedding Generator, we apply the module to state-of-the-art model STTR (Li et al., 2021) targeting at stereo matching, where model structure is similar to ours. The results are shown in Table 1.

4.2 Relative Pose Estimation

The accuracy of camera pose estimated from the corresponding points is typically used to evaluate the quality of feature correspondences. The evaluation protocol follows SuperGlue (Sarlin et al., 2020), where feature matches generated are used to solve essential matrix with RANSAC. We report the AUC of pose error at thresholds of $5°, 10°, 20°$, and pose error is defined as the maximum of angular error in rotation and translation following Sarlin et al. (2020). Camera extrinsic is used to construct 3D Position Embedding in PA-LoFTR which greatly improve the quality of feature matching. In relative pose estimation task, camera extrinsic is unavailable and we propose a pose-refine mechanism to fit PA-LoFTR. We add an additional simplified LoFTR branch that generates a pose estimation first, and then PA-LoFTR constructs the 3D position embedding to get final matches. The details of implementation can be found in supplementary materials. The results in Table 2 show that PA-LoFTR with pose-refine mechanism can improve the estimated pose by a large margin compared to LoFTR under the same training setting.



(a) SP+SuperGlue (b) LoFTR† (c) PA-LoFTR† (pose refine) (d) PA-LoFTR†

Figure 4: **Qualitative Results.** PA-LoFTR is compared to LoFTR (Sun et al., 2021) and SuperGlue (Sarlin et al., 2020) on indoor dataset. The proposed pose-refine PA-LoFTR can provide more correct matches and less false matches compared to SuperGlue and LoFTR under challenging cases. PA-LoFTR can generate even better results where ground truth relative pose is provided to the model. The red color means epipolar error of the match is beyond 5×10^{-4}.

We conduct the experiments with the lite training setting as described in section 3.5 to show effectiveness of PA-LoFTR. As shown in Table 2, PA-LoFTR can improve the performance of pose estimation compared to LoFTR under the same training settings. It is noticed that out method can provide accurate matches on testing dataset even trained with limited splits of ScanNet, which shows its ability of generalization. It is expected that PA-LoFTR can have even better performance if we can train the model following the setting of original LoFTR, where all 1513 scenes of ScanNet are used for training and 64 GPUs are used with a batch size of 64. We also test the case where

Method	**ScanNet**		
	auc@5°	auc@10°	auc@20°
ORB+GMS (Bian et al., 2017)	5.21	13.65	25.36
SP+SuperGlue (Sarlin et al., 2020)	16.16	33.81	51.84
LoFTR* (Sun et al., 2021))	20.06	40.8	57.62
LoFTR†	13.08	27.42	43.41
PA-LoFTR†	**27.74**	**47.35**	**64.19**
PA-LoFTR† (pose refine)	17.04	34.28	51.18

Table 2: **Relative Pose Estimation on ScanNet dataset.** The AUC of pose error is reported, which reflects quality of feature matching detected. PA-LoFTR can improve matching performance a lot. Without camera extrinsic, PA-LoFTR can also achieve promising results compared with state-of-the-art methods. $*$ means full training setting in LoFTR with 64 GPUs and a batch size of 64 on full ScanNet dataset. † means our lite training setting on partial ScanNet where 4 GPUs are used with a batch size of 8 as described in section 3.5. Models are tested on consistent dataset following SuperGlue (Sarlin et al., 2020).

ground truth pose is available, and PA-LoFTR can show its full power and give high-quality feature matches. We expect that in tasks where camera extrinsic is available, 3D position embedding can provide stronger prior knowledge to the model and PA-LoFTR can show more effectiveness.

4.3 Ablation Study

Compared to LoFTR, the main module is the 3D Position Embedding Generator which bridges the gap between camera extrinsic and feature embedding for Transformer. We evaluate 4 variant settings of 3D Position Embedding Generator as in Table 3: 1) Apply only rotation matrix of camera intrinsic to generator position embeddings, which results in a drop of accuracy. 2) Apply position embeddings once at the beginning of Transformer Encoder. 3) Apply position embeddings for self-attention layers in Transformer Encoder. We find decreasing the use of 3D position embedding in

Method	ScanNet		
	auc@5°	auc@10°	auc@20°
PA-LoFTR	27.74	47.35	64.19
$\boldsymbol{R}$ matrix for 3D PE	23.21	42.98	58.67
One-time position embedding	24.91	43.40	59.87
Position embedding for self-attention layers	26.10	44.82	61.08
Estimated pose for position embedding	17.04	34.28	51.18
Without Depth Feature Generator	28.63	48.42	65.88
Depth feature for 3D position embedding	21.08	39.79	56.98
Position embedding for the first cross attention	27.16	46.69	63.56

Table 3: **Ablation Study.** Four variants of 3D PE Generator and three variants of usage of depth features by Depth Feature Generator. All models are trained under lite setting † as described in Table 2.

Transformer layers will both decrease accuracy. 4) Apply estimated pose with pre-trained LoFTR to generator 3D postion embeddings instead of using ground truth extrinsic, which is essential for pose estimation task. The usage of estimated pose will cause decrease of matching performance compared to full PA-LoFTR, but can explicitly refine the estimated pose.

We also evaluate 3 variant settings related to usage of depth feature generator: 1) Do not apply depth feature generator. While it leads to decrease in performance on validation dataset compared to full PA-LoFTR (about 3 points decrease), the model performs better on testing dataset. The Depth Feature Generator may have limited ability of generalization. It is expected to have better performance if trained with more scenes. 2) Apply depth feature in 3D PE generator, where we use depth probability to guide generation of 3D PE, which does not give better performance. 3) Apply 3D position embedding at the first cross attention layer between depth features and visual features, which does not improve performance compared to giving not 3D PE to the first cross attention.

5 Conclusion

We presents a novel feature matching approach, PA-LoFTR, which introduces 3D position embeddings to Transformer encoder to bridge the gap between 3D position information and 2D image features. The proposed method embeds the points cloud for each image as position features and applies self and cross attention layers to transform image features with 3D space information. PA-LoFTR can detect precise and plenty of feature matches even under challenging pose changes. The experiments show that PA-LoFTR can reach state-of-the-art performance, and we expect PA-LoFTR can be extended to more tasks in which camera extrinsic is known, where PA-LoFTR can show its full power.

REFERENCES

Abhishek Badki, Alejandro Troccoli, Kihwan Kim, Jan Kautz, Pradeep Sen, and Orazio Gallo. Bi3D: Stereo depth estimation via binary classifications. In *arXiv preprint arXiv:2005.07274*, 2020.

J. Bian, W. Y. Lin, Y. Matsushita, S. K. Yeung, and M. M. Cheng. Gms: Grid-based motion statistics for fast, ultra-robust feature correspondence. In *IEEE Conference on Computer Vision & Pattern Recognition*, 2017.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. 2020.

Yilun Chen, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Dsgn: Deep stereo geometry network for 3d object detection. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12533–12542, 2020.

A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and Matthias Niener. Scannet: Richly-annotated 3d reconstructions of indoor scenes. 2017.

D. Detone, T. Malisiewicz, and A. Rabinovich. Toward geometric deep slam. 2017.

Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPR Deep Learning for Visual SLAM Workshop*, 2018.

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 *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

U. Efe, K Gokalp Ince, and A Aydin Alatan. Dfm: A performance baseline for deep feature matching. *arXiv e-prints*, 2021.

Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. In Martin A. Fischler and Oscar Firschein (eds.), *Readings in Computer Vision*, pp. 726–740. Morgan Kaufmann, San Francisco (CA), 1987. ISBN 978-0-08-051581-6.

Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

Xiaoyang Guo, Kai Yang, Wukui Yang, Xiaogang Wang, and Hongsheng Li. Group-wise correlation stereo network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3273–3282, 2019.

Sunghwan Hong, Seokju Cho, Seungryong Kim, and Stephen Lin. Integrative feature and cost aggregation with transformers for dense correspondence, 2022.

K. C. Huang, T. H. Wu, H. T. Su, and W. H. Hsu. Monodtr: Monocular 3d object detection with depth-aware transformer. 2022.

Achin Jain, Kibok Lee, Gurumurthy Swaminathan, Hao Yang, Bernt Schiele, Avinash Ravichandran, and Onkar Dabeer. Completr: Reducing the cost of annotations for object detection in dense scenes with vision transformers, 2022.

Wei Jiang, Eduard Trulls, Jan Hosang, Andrea Tagliasacchi, and Kwang Moo Yi. COTR: Correspondence Transformer for Matching Across Images. 2021.

A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. 2020.

Xinghui Li, Kai Han, Shuda Li, and Victor Prisacariu. Dual-resolution correspondence networks. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

Zhaoshuo Li, Xingtong Liu, Nathan Drenkow, Andy Ding, Francis X. Creighton, Russell H. Taylor, and Mathias Unberath. Revisiting stereo depth estimation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6197–6206, October 2021.

Zhikai Li, Mengjuan Chen, Junrui Xiao, and Qingyi Gu. Psaq-vit v2: Towards accurate and general data-free quantization for vision transformers, 2022.

T. Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017a.

T. Y. Lin, P. Goyal, R. Girshick, K. He, and P Dollár. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, PP(99):2999–3007, 2017b.

Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr: Position embedding transformation for multi-view 3d object detection. *arXiv preprint arXiv:2203.05625*, 2022.

D. G. Low. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2004.

D. G. Lowe. Lowe, d.g.: Distinctive image features from scale-invariant key-points. int. j. comput. vision 60(2), 91-110. *International Journal of Computer Vision*, 60(2), 2004.

Runyu Mao, Chen Bai, Yatong An, Fengqing Zhu, and Cheng Lu. 3dg-stfm: 3d geometric guided student-teacher feature matching, 2022.

Kevin Miao, Akash Gokul, Raghav Singh, Suzanne Petryk, Joseph Gonzalez, Kurt Keutzer, Trevor Darrell, and Colorado Reed. Prior knowledge-guided attention in self-supervised vision transformers, 2022.

Guocheng Qian, Xingdi Zhang, Abdullah Hamdi, and Bernard Ghanem. Pix4point: Image pretrained transformers for 3d point cloud understanding, 2022.

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

I. Rocco, R. Arandjelović, and J. Sivic. Efficient neighbourhood consensus networks via submanifold sparse convolutions. *arXiv preprint arXiv:2004.10566*, 2020.

Ignacio Rocco, Mircea Cimpoi, Relja Arandjelović, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Ncnet: Neighbourhood consensus networks for estimating image correspondences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(2):1020–1034, 2022. doi: 10.1109/TPAMI.2020.3016711.

E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski. Orb: an efficient alternative to sift or surf. In *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011*, 2011.

Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *CVPR*, 2020.

T. Sattler, B. Leibe, and L. Kobbelt. Scramsac: Improving ransac's efficiency with a spatial consistency filter. In *Computer Vision, 2009 IEEE 12th International Conference on*, 2009.

Jiaming Sun, Linghao Chen, Yiming Xie, Siyu Zhang, Qinhong Jiang, Xiaowei Zhou, and Hujun Bao. Disp r-cnn: Stereo 3d object detection via shape prior guided instance disparity estimation. In *CVPR*, 2020.

Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-free local feature matching with transformers. *CVPR*, 2021.

Shitao Tang, Jiahui Zhang, Siyu Zhu, and Ping Tan. Quadtree attention for vision transformers. In *International Conference on Learning Representations*, 2022.

P. Truong, M. Danelljan, and R. Timofte. Glu-net: Global-local universal network for dense flow and correspondences. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

T. Tuytelaars and L. V. Gool. Wide baseline stereo matching based on local, affinely invariant regions. 2000.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *arXiv*, 2017.

Yue Wang, Vitor Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, , and Justin M. Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *The Conference on Robot Learning (CoRL)*, 2021.

K. M. Yi, E. Trulls, V. Lepetit, and P. Fua. Lift: Learned invariant feature transform. *Springer International Publishing*, 2016.

Feihu Zhang, Victor Prisacariu, Ruigang Yang, and Philip HS Torr. Ga-net: Guided aggregation net for end-to-end stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 185–194, 2019.

R. Zhang, H. Qiu, T. Wang, X. Xu, Z. Guo, Y. Qiao, P. Gao, and H. Li. Monodetr: Depth-aware transformer for monocular 3d object detection. 2022.

Qunjie Zhou, Torsten Sattler, and Laura Leal-Taixe. Patch2pix: Epipolar-guided pixel-level correspondences. In *CVPR*, 2021.

Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.