IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 26 December 2024, accepted 15 January 2025, date of publication 20 January 2025, date of current version 30 January 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3531896

RESEARCH ARTICLE

Camera Absolute Pose Estimation Using Hierarchical Attention in Multi-Scene

XINHUA LU[1], JINGUI MIAO[2], QINGJI XUE[1], HUI WAN[2], AND HAO ZHANG[2]
[1]Electronic Information Institute, Nanyang Institute of Technology, Nanyang 473004, China
[2]School of Computer and Artificial Intelligence, Zhengzhou University, Zhengzhou 450001, China

Corresponding author: Xinhua Lu (ieluxinhua@sina.com)

The research for this article was supported in part by the Natural Science Foundation of Henan under Grant 222300420504, in part by the Academic Degrees and Graduate Education Reform Project of Henan Province under Grant 2021SJGLX262Y, in part by the Postgraduate Education Reform and Quality Improvement Project of Henan Province under Grant YJS2024AL141, and in part by the Foundation and Frontier Project of Nanyang under Grant JCQY008.

ABSTRACT The multi-scene camera pose estimation approach aims to recover the camera pose from any given scene, catering to the demands of real-life mobile devices to perform tasks. Facing the challenge that it is difficult to extract efficient features in training multi-scene models, we present a modified model named Hierarchical Attention Absolute Pose Regression(H-AttnAPR) which can obtain different scales of feature dependencies. A Hierarchical Attention(HA) module is introduced prior to the scene classification module, where it captures both intra- and inter-correlations among image patches, utilizing both local and global key information from images to restore the absolute camera pose without the need for additional point cloud data. H-AttnAPR efficiently models global dependencies without compromising fine-grained feature information. Therefore, it overcomes the limitations that solely focus on long-range pixel-level feature dependencies within images while neglecting local patches of image feature dependencies. Our approach has been validated on the 7Scenes and Cambridge benchmark datasets. Compared to the baseline algorithm PoseNet, our algorithm has achieved a 41.1% reduction in translation error and a 61.1% decrease in rotation error, demonstrating superior performance in multi-scene absolute camera pose regression.

INDEX TERMS Image processing, pose estimation, attention mechanisms, feature extraction, deep learning.

I. INTRODUCTION

In this section, we delve into the realm of camera pose estimation techniques, offering a comprehensive overview. We then pivot to explore the existing landscape of related research in Section II. Section III is dedicated to detailing the architecture of our proposed system. Section IV delves into an experimental analysis to validate our approach. Finally, we conclude our contributions in Section V, summarizing the essence of our work.

Visual localization focuses on recovering the spatial position of camera, utilizing a single or a sequence of images. As Deep Learning(DL) and Computer Vision(CV) technologies [1] have flourished in recent years, camera pose estimation has garnered extensive attention in both academic and industrial communities [2], [3]. Accurate and robust camera pose estimation is pivotal for downstream tasks such as object localization [4], intelligent obstacle avoidance [5], automation and robotics [6]. However, due to the complexity arising from factors such as varying viewpoints, lighting conditions, or dynamic variation related to both the cameras and the environments being captured, this task remains highly chanllenging. Current methods for camera pose estimation offer varying trade-offs between accuracy, speed, and computational complexity. For instance, methods that rely on structural information [2], [3], [7] achieve state-of-the-art(SOTA) localization accuracy, however, their inference time can reach several hundred milliseconds, posing difficulties in deployment on lightweight devices. Absolute Pose Regression methods (APRs) [8] directly compute the camera pose based solely on the provided query image. APRs require minimal memory and fast inference speed, completing the process in just tens of milliseconds. They have the potential to be deployed on edge devices that

The associate editor coordinating the review of this manuscript and approving it for publication was Kumaradevan Punithakumar.

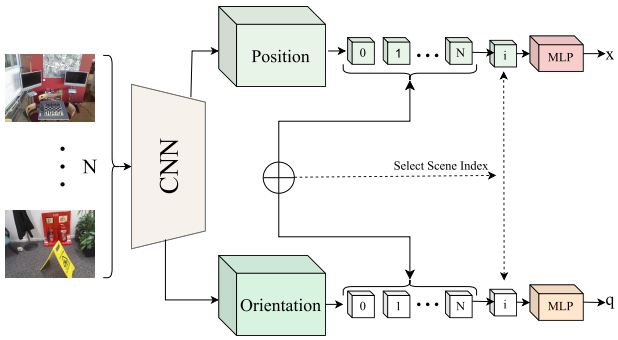


FIGURE 1. A simplified schematic for multi-scene absolute pose estimation. The diagram features two branches, both integrating HA and transformers, which are used to process feature maps obtained from the convolutional backbone for regressing camera position and rotation, respectively. For scene recognition, the two branches are merged to determine the scene. The selected scene output is then used to regress the position (x) and rotation (q).

have constrained computing resources. However, compared with structure-based methods [7], the accuracy of APRs is typically one order of magnitude lower [9].

Absolute Pose Regression(APR) was originally formulated by Kendall et al. [8], following the remarkable achievements demonstrated by Convolutional Neural Network(CNN). They utilized a CNN backbone and a Multi-Layer Perceptron (MLP) as the regression module for pose estimation. PoseNet offered an efficient and minimalistic approach for estimating the pose quickly. While it may not match the accuracy of matching-based algorithms, it opened new avenues for camera pose prediction, leading to numerous subsequent studies that aimed to improve localization accuracy and generalization performance through modifications to the backbone network and MLP architecture [10], [11], cost functions [12], and optimization algorithms [13], [14]. Most of the APRs are designed to train a dedicated model for each individual scene, and considering a comprehensive dataset comprising N scenarios, the challenge lies in training N models and deploying them, which will greatly increase the complexity of model application. These characteristics reveal the limitations of single-scene APR for practical applications.

Blanton et al. [15] extended the paradigm of single-scene training models to multiple scenes, but they still need to train models with multiple scenarios in the training stage, and then select the appropriate ones. Shavit et al. [16] further explored the advantages of self-attention(SA) in image processing tasks. They mainly used SA to focus on the global dependence of image pixels. Their team then put forward a camera pose auto-encoder scheme [17], utilizing work [16] to train an absolute pose auto-encoder. This method introduced prior information to improve the positioning accuracy, but the performance of the auto-encoder depended largely on the performance of the APR used as a teacher, and the two models need to be trained separately to regress the pose. The above works focus on abstract global pixel-level feature dependencies in images, but failed to obtain the local dependencies within the image blocks under multi-scene.

Inspired by [18] which introduced the HA mechanism, we made improvements to the model of [16], the later only focuses on global attention using SA. In our approach, we specifically consider not only the connections inside local patches within images but also the dependencies between global pixels to enhance the model's ability to capture fine-grained details. We send the feature map obtained from the CNN backbone into the HA module, and obtain the local and global information weights of the image after block re-aggregation processing, and generate effective features with rich granularity by aggregating features from hierarchical representations of global and local features. Our method is evaluated on the Cambridge Landmarks and 7Scenes datasets. Experiments demonstrate that our method achieves greater performance in terms of camera pose estimation accuracy with multi-scene training.

In summary, our key contributions encompass the following:

- We design a multi-scene camera absolute pose estimation network named H-AttnAPR, enhancing the accuracy of camera localization. We employ a HA module following CNN to capture the dependencies between local and global within the feature map, demonstrating the advantages of HA in image feature representation.
- We improve the scene classification module and the pose regression module and set the model parameters according to the respective characteristics of the two branches.
- We achieve the most advanced localization accuracy of our known multi-scene APRs on both indoor and outdoor benchmarks.

II. RELATED WORK

In this chapter, we conduct an in-depth examination and analysis of the research in the relevant field. We start by exploring the structure-based camera pose estimation techniques. Following this, we investigate the regression-based approach to camera pose estimation, evaluating its strengths and limitations, as well as its performance across different scenarios. Table1 summarizes the three mainstream technologies. Building on this, we propose our method and compare it with the previously discussed techniques to highlight the advantages and innovations of our proposed approach in the task of camera pose estimation.

A. STRUCTURE-BASED CAMERA POSE ESTIMATION

The localized pipeline based on structural refers to the method of recovering camera's position and orientation relies on the successful matching of features in the query image to the three-dimensional(3D) architectural features of the scene's representation. This process involves utilizing Structure from Motion (SFM) [19] for the construction of 3D

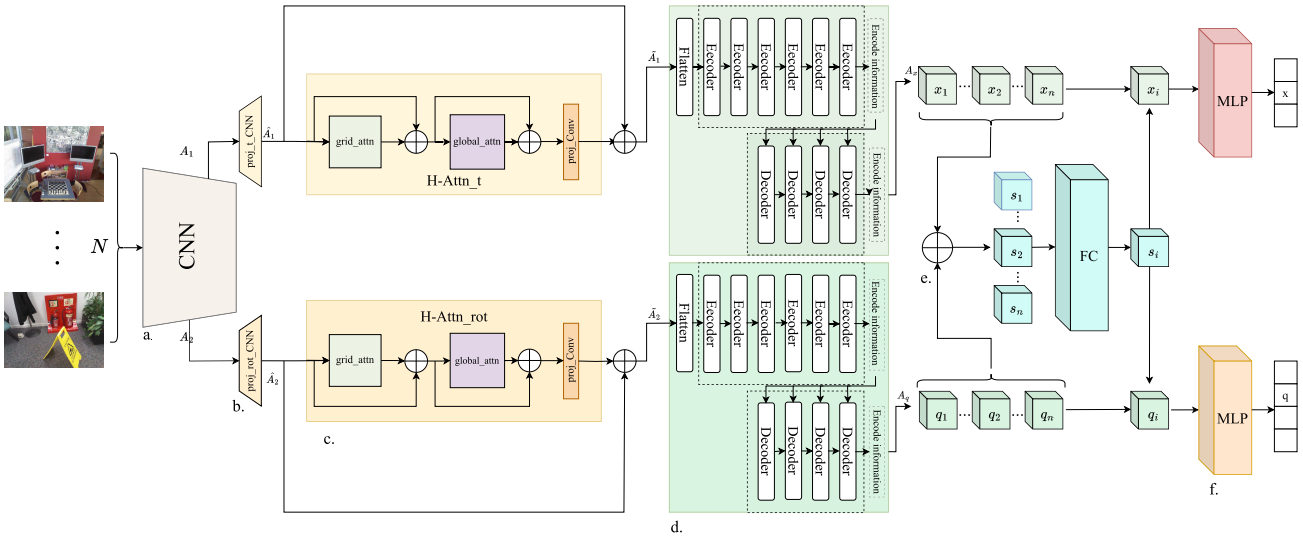


FIGURE 2. The system architecture of our proposed. There are two branches, which are translation and rotation, respectively. N images of different scenes are input into the model. The feature maps from the CNN backbone network first pass through an HA module, and then are fed into the scene classification module. Finally, the encode features are utilized to regress the camera pose(x and q).

TABLE 1. A comparative study on camera pose estimation techniques.

Method	The technology used	Merits	Limitations
Structure-based [19] [7]	Feature matching, SFM, 3D reconstruction, PnP, RANSAC, IR	SOTA positioning accuracy	High computation time, training is cumbersome and difficult to deploy.
RPRs [20]–[22]	IR, interpolation, database retrieval	Higher accuracy in small-scale indoor benchmarks, improved generalization capabilities	Error accumulation, high initialization requirements, long inference time.
APRs [8] [15] [16]	CNN, FC,Attention Mechanism	A single forward pass, fast, multi-scene training	A loss in the accuracy of pose estimation.

point cloud model that captures the entire scene structure. Compared with the camera pose regression methods that solely rely on object features in the image, the pipeline based on structural features relies more heavily on the prior information of the 3D scene model. DSAC [7] proposed by Brachmann et al. adopts a CNN architecture to predict the 3D positions of pixels in the queried image, allowing for the alignment of 3D point clouds from the scene model with the image. Afterwards, they employ Perspective-n-Point (PnP) algorithm to calculate the position of camera and use Random Sample Consensus (RANSAC) method to remove outliers [12]. The above methods have achieved SOTA positioning accuracy in the two-step scheme of image retrieval and feature matching, but they require a relatively high computation time and necessitate training individual models for each scene, making it difficult to deploy in practical applications.

B. REGRESSION-BASED CAMERA POSE ESTIMATION

Regression based camera pose estimation techniques are categorized into relative camera pose regression methods(RPRs) and APRs.

RPRs combine camera pose regression with image retrieval(IR), which is used to identify nearby images, and then evaluate the relative motion between the query and all the retrieved images separately, followed by pose interpolation. Therefore, the learning focuses on regressing the relative pose given a pair of images [20], [21], [22]. RPRs have shown improved generalization capabilities over APRs, resulting in higher accuracy in small-scale indoor benchmarks [22]. Nevertheless, just as other IR-based techniques, this method requires an initial retrieval stage and a database containing pose labels in the phase of inference. Therefore, it takes more time in the inference phase, and no research has yet proved its equal effectiveness in large-scale benchmark tests.

APRs originated from the work [8]. Although it is not as accurate as structure-based methods, it can estimate poses using a single forward pass, offering the advantages of being lightweight and fast. In order to break the limitation of the scenario-specific model, Blanton et al. extended the single-scene APR to multi-scene and proposed MSPN, which was a multi-scene absolute pose regression method [15]. It used a SoftMax fully connected(FC) layer to predict the scenario. For each scene, a separate FC layer was

trained specifically. However, the proposed framework still involves training multiple models (one for each scenario) and then adopting a hybrid expert-guided [23] method to select the most appropriate model. Shavit et al. [16] applied a transformer model to predict the scene for each image and the obtained feature representation was then passed on an MLP head for regressing camera pose, training only one model to handle different scenes. Shavit team then extended their previous work by adding a pose auto-encoder (PAE) [17] to the MS-Transformer. It was trained through a teacher-student method, where a pre-trained APR served as a teacher that can regress the location information of the camera from the image, and the student PAE, as an MLP, learnt to encode the camera pose into the latent representation learned by APR from the corresponding image. This encoding served as prior information to guide APR to achieving accurate pose estimation. However, the accuracy of the trained PAE is affected by the APR. If there are errors in the APR, these errors will be transmitted to the PAE, thereby increasing the error of pose regression.

There are some limitations inherent in the previous research. MS-Transformer and Auto-encoder mainly use self-attention to solve the long-distance dependence between image pixels and predict the scene of the image. While they overlook the feature dependence within local image blocks.

C. OUR METHODS

The paradigm of multi-scene pose estimation as shown in Figure1, includes feature extraction, scene classification, and pose regression. The model's efficient feature extraction ability is important for this task. In our work, considering the importance of the representation of key features in the image for predicting pose in multi-scene, we pay attention to the dependencies within image blocks rather than merely the individual pixels. Based on the framework [16], we employ a HA [18] module to first segment the feature maps from the CNN backbone into blocks to calculate local feature dependencies, and merge these blocks to obtain global dependencies. Then we aggregate the two representations and achieve a more fine-grained feature representation. Since we divide the image into patches to process, the inference process remains swift and efficient, with minimal time required, making it favorable for real-time applications. Besides, our network architecture only requires one model for dataset training. Compared with the known leading architectures in this field, we have achieved greater pose estimation performance.

III. SYSTEM ARCHITECTURE

The approach to estimating camera pose across multiple scenes generally involves using a training set that includes all scenarios to train a single network model that can adapt to different scenes. Building upon the framework established in reference [16], this paper introduces an enhanced approach that integrates the technique from [18] into the existing schema. We name it H-Attn APR and use HA to capture and represent the local and global dependencies within images that processed by a CNN backbone. Camera pose can be represented as a tuple $\mathbf{p} = \langle \mathbf{x}, \mathbf{q} \rangle$. The pose $\mathbf{p}$ is defined relative to an arbitrary global coordinate system, where $\mathbf{x} \in \mathbb{R}^3$ represents the spatial position of the camera within the global coordinate domain, and $\mathbf{q} \in \mathbb{R}^4$ represents camera's orientation in the form of a quaternion [8].

A. SYSTEM PARADIGM

Our network, as illustrated in Fig2, takes N images $\mathbf{I} \in \mathbb{R}^{h \times w \times c}$ of scenes as input. Initially as shown in step (a.) of Figure 2, the model performs downsampling operations through a CNN backbone to obtain feature maps $\mathbf{A_1} \in \mathbb{R}^{c_1 \times h_1 \times w_1}$ and $\mathbf{A_2} \in \mathbb{R}^{c_2 \times h_2 \times w_2}$.

Then, $\mathbf{A_1}$ and $\mathbf{A_2}$ are reprojected through a 2-Dimensional convolution to suitable dimensional representations $\hat{\mathbf{A_1}} \in \mathbb{R}^{c \times h_1 \times w_1}$ and $\hat{\mathbf{A_2}} \in \mathbb{R}^{c \times h_2 \times w_2}$ for HA. The purpose of this operation is to standardize the feature maps obtained from the CNN network to achieve consistent output. After that, HA is applied as shown in step (c.) of Figure 2 to obtain local and global attention representations:

$$\mathbf{HAttn_t} \in \mathbb{R}^{c \times h_1 \times w_1} = \mathbf{HA}(\hat{\mathbf{A_1}}), \tag{1}$$

$$\mathbf{HAttn_{rot}} \in \mathbb{R}^{c \times h_2 \times w_2} = \mathbf{HA}(\hat{\mathbf{A_2}}). \tag{2}$$

Then add to $\hat{\mathbf{A_1}}$ and $\hat{\mathbf{A_2}}$:

$$\tilde{\mathbf{A_1}} \in \mathbb{R}^{c \times h_1 \times w_1} = \mathbf{HAttn_t} + \hat{\mathbf{A_1}}, \tag{3}$$

$$\tilde{\mathbf{A_2}} \in \mathbb{R}^{c \times h_2 \times w_2} = \mathbf{HAttn_{rot}} + \hat{\mathbf{A_2}}, \tag{4}$$

and we obtained activation maps with attention weights for key image features. Subsequently, we input the activation map into the improved transformer architecture based on [16]. The transformer decoder outputs two query sequences $\{\mathbf{x_i}\}_1^N$ and $\{\mathbf{q_i}\}_1^N$, each carrying scene embedding vectors. Since each input image corresponds to only one scene, we need to identify the scene that matches the image. As shown in step (e.) of Figure 2, we append $\{\mathbf{x_i}\}_1^N$ and $\{\mathbf{q_i}\}_1^N$ as $\{\mathbf{s_i}\}_1^N$

$$\mathbf{s_i} = \begin{bmatrix} \mathbf{x_i} \\ \mathbf{q_i} \end{bmatrix} \in \mathbb{R}^{2c} \tag{5}$$

and process them through a fully connected layer to determine the scene index with the highest probability. The determined outputs $\{\mathbf{x_i}, \mathbf{q_i}\}$ from the Transformer are then fed into a MLP with a single hidden layer, as shown in Figure 2 (f.), to regress the position and orientation vectors $\mathbf{x} \in \mathbb{R}^3$ and $\mathbf{q} \in \mathbb{R}^4$.

B. HIERARCHICAL ATTENTION

The detailed implementation of step (c.) in Figure 2 is shown in Figure 3, the HA mechanism, in training the camera pose estimation model for multiple scenes, achieves the modeling of global dependencies and the preservation of local details by combining both downsampling [24] and window-based [25] transformers. This mechanism enables the model to capture the global structure of scenes, understand the overall layout,

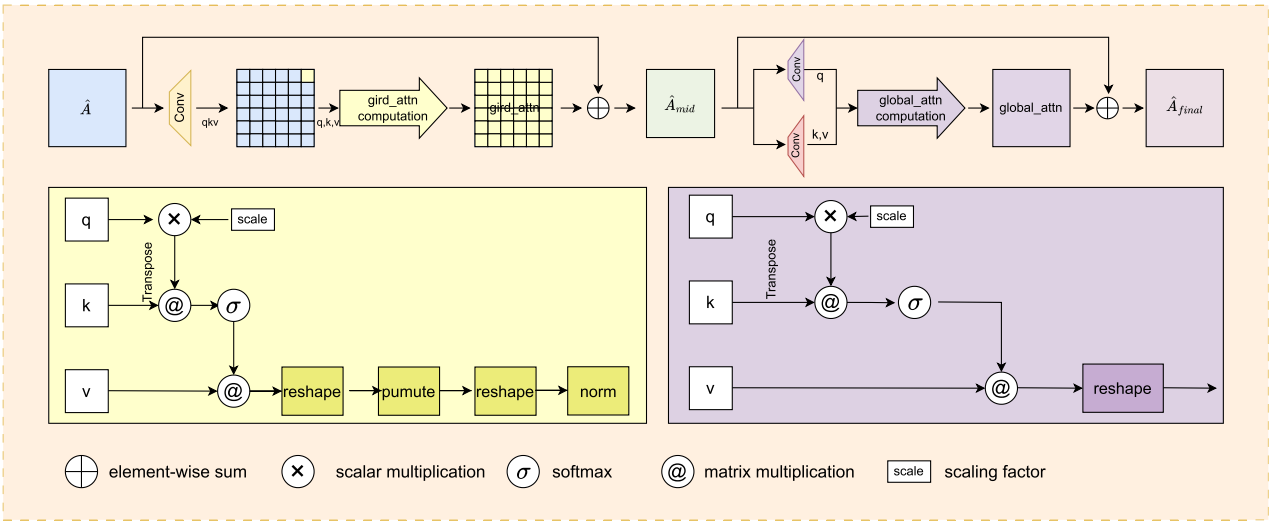


FIGURE 3. HA module. The yellow box contains the computation for grid attention, while the purple box encompasses the computation for global attention.

and consistently estimate camera pose across various scenes. Additionally, it focuses on local features such as edges and corners, which are vital cues for precise camera pose estimation, thereby ensuring that no important details are overlooked during the downsampling process. The capability to process information at multiple scales allows the model to consider both the macroscopic structure and the microscopic details of scenes, integrating information from different levels to form a comprehensive understanding. This enhances the model's ability to interpret complex scenes, leading to more accurate and reliable camera pose estimation.

In the initial stage of computing local attention for the feature map, the initial input originates from the feature map $\mathbf{X} = \mathbb{R}^{H\times W\times C}$ processed by a CNN backbone. The feature map is first divided into smaller grids of size $G_1 \times G_1$, and the original feature map is reshaped as follows:

$$\begin{aligned} &\mathbf{X} \in \mathbb{R}^{H\times W\times C} \\ \rightarrow\ &\mathbf{X_1} \in \mathbb{R}^{\left(\frac{H}{G_1}\times G_1\right)\times\left(\frac{W}{G_1}\times G_1\right)\times C}, \qquad (6) \\ \rightarrow\ &\mathbf{X_1} \in \mathbb{R}^{\left(\frac{H}{G_1}\times\frac{W}{G_1}\right)\times(G_1\times G_1)\times C}. \qquad (7) \end{aligned}$$

The $\mathbf{Q_1}$, $\mathbf{K_1}$ and $\mathbf{V_1}$ are then computed using the following equations:

$$\mathbf{Q_1} = \mathbf{X_1}\mathbf{W_1^q}, \mathbf{K_1} = \mathbf{X_1}\mathbf{W_1^k}, \mathbf{V_1} = \mathbf{X_1}\mathbf{W_1^v}, \qquad (8)$$

where $\mathbf{W_1^q}, \mathbf{W_1^k}, \mathbf{W_1^v} \in \mathbb{R}^{C\times C}$ are the trained weight matrices. After this, the following operation is applied:

$$\mathbf{A} = Softmax(QK^T/\sqrt{d})V. \qquad (9)$$

This yields a local attention map $\mathbf{A_1}$, To facilitate network optimization, $\mathbf{A_1}$ is reshaped to match the shape of $\mathbf{X}$:

$$\begin{aligned} &\mathbf{A_1} \in \mathbb{R}^{(\frac{H}{G_1}\times\frac{W}{G_1})\times(G_1\times G_1)\times C} \\ \rightarrow\ &\mathbf{A_1} \in \mathbb{R}^{(\frac{H}{G_1}\times G_1)\times(\frac{W}{G_1}\times G_1)\times C}, \qquad (10) \\ \rightarrow\ &\mathbf{A_1} \in \mathbb{R}^{H\times W\times C}. \qquad (11) \end{aligned}$$

Finally, a residual connection is incorporated:

$$\mathbf{A_1} = \mathbf{A_1} + \mathbf{X}. \qquad (12)$$

This obtains local attention maps for each grid of size $G_1 \times G_1$.

Subsequently, we calculate the global attention for the feature map. During the computation of the key and value matrices, $\mathbf{A_1}$ is downsampled by a factor of G_2 to make the calculation more efficient. We consider each $G_2 \times G_2$ block as a token. This process can be formulated as:

$$\hat{\mathbf{A}}_\mathbf{1} = AvePool_{G_2}(\mathbf{A_1}), \qquad (13)$$

where $AvePool_{G_2}(\cdot)$ refers to the application of average pooling to a feature map with kernel size and stride both set to G_2, performing G_2 times downsampling. Then we get $\hat{\mathbf{A}}_\mathbf{1} \in \mathbb{R}^{\frac{H}{G_2}\times\frac{W}{G_2}\times C}$. Following that $\mathbf{A_1}$ and $\hat{\mathbf{A}}_\mathbf{1}$ are reshaped as follows:

$$\mathbf{A_1} \in \mathbb{R}^{H\times W\times C} \rightarrow \mathbf{A_1} \in \mathbb{R}^{(H\times W)\times C}, \qquad (14)$$

$$\hat{\mathbf{A}}_\mathbf{1} \in \mathbb{R}^{\frac{H}{G_2}\times\frac{W}{G_2}\times C} \rightarrow \hat{\mathbf{A}}_\mathbf{1} \in \mathbb{R}^{(\frac{H}{G_2}\times\frac{W}{G_2})\times C}. \qquad (15)$$

After this, we compute the queries, keys, and values using the following equations:

$$\mathbf{Q_2} = \mathbf{A_1}\mathbf{W_2^q}, \mathbf{K_2} = \mathbf{A_1}\hat{W}_2^k, \mathbf{V_2} = \mathbf{A_1}\hat{W}_2^v, \qquad (16)$$

where $\mathbf{W_2^q}, \mathbf{W_2^k}, \mathbf{W_2^v} \in \mathbb{R}^{C\times C}$ are trainable weight matrices. From the above equations, it is easy to deduce that $\mathbf{Q_2} \in$

$\mathbb{R}^{(H\times W)\times C}$, $\mathbf{V_2} \in \mathbb{R}^{(\frac{H}{G_2}\times\frac{W}{G_2})\times C}$. Then, we apply the attention score calculation Eq.9 to obtain the global attention feature map $\mathbf{A_2} \in \mathbb{R}^{(H\times W)\times C}$ and reshape it as:

$$\mathbf{A_2} \in \mathbb{R}^{(H\times W)\times C} \rightarrow \mathbf{A_2} \in \mathbb{R}^{H\times W\times C}. \tag{17}$$

The final output of this HA is given by:

$$HAttn(\mathbf{X}) = (\mathbf{A_1} + \mathbf{A_2})\mathbf{W^p} + \mathbf{X}, \tag{18}$$

where $\mathbf{W_p} \in \mathbb{R}^{C\times C}$ is a weight matrix used for feature reprojection.

The feature maps processed by HA mechanism significantly enhance the representation of the feature maps by precisely highlighting key information in the images, such as edges and corners, while effectively suppressing irrelevant noise, thereby increasing the concentration and relevance of the information. Furthermore, this mechanism achieves multi-scale information fusion, which not only enhances the model's grasp of the global structure but also strengthens its understanding of local details, thus deepening the comprehensive interpretation of the scene. At the same time, the robustness of the model is also improved, enabling it to more flexibly handle scene changes and occlusion issues. Notably, the HA also reduces the computational burden and optimizes the efficiency of resource usage. It greatly enhances the model's generalization ability, ensuring good transferability across different scenes.

C. LOSS FUNCTION

The positioning error of the camera pose is calculated by measuring the difference between the estimated pose and the ground truth pose, represented by $\mathbf{p_{gt}} = \langle \mathbf{x_{gt}}, \mathbf{q_{gt}} \rangle$. Our model is trained to minimize the translation error L_x and the rotation error L_q, which are defined as follows:

$$L_x = ||\mathbf{x_{gt}} - \mathbf{x}||_2, \tag{19}$$

$$L_q = ||\mathbf{q_{gt}} - \frac{\mathbf{q}}{||\mathbf{q}||}||_2. \tag{20}$$

Here, $\mathbf{q}$ is normalized to a unit quaternion to ensure a valid representation. We adopt the formulation suggested by Shavit et al. [16] for the comprehensive loss function of our model:

$$L_{MS} = L_p + NLL(s, s_{gt}), \tag{21}$$

where L_p is a pose error loss formulated by Kendall et al. [12] using a Laplacian-like likelihood for both translation and rotation errors:

$$L_p = L_x exp(-s_x) + s_x + L_q exp(-s_q) + s_q, \tag{22}$$

where s_x and s_q are the learned parameters that regulate the trade-off between the loss of L_x and L_q. Since we are training on multiple scenes and need to match the corresponding scenes from the training images, we utilize the negative log-likelihood loss (NLL) for scene classification. This loss aims to accurately categorize the scenes during training.

IV. EXPERIMENTAL ANALYSIS

This section describes the model configuration, datasets, comparison against existing methods, ablation studies and the evaluation of noise resistance performance.

A. CONFIGURATION

We utilize the Adam algorithm to tune our model's parameters, thereby ensuring that the loss function specified in Eq.21 is minimized, with the settings of $s_x = 0.0$ and $s_q = -3.0$, following the design principles outlined in [12]. Taking into account the model performance as well as considerations of time and memory, we choose EfficientNetB0 as the CNN backbone network and our training is based on the pre-trained model efficient-net-b0.pth. The batch size is configured to 12 with a learning rate of 10^{-4}. We set the step decay as 20. Due to the differences between indoor and outdoor scenes, we set $attn_t = 2$ and $attn_{rot=4}$ for the 7Scenes dataset, while to the Cambridge Landmark dataset, we use $attn_t = 4$ and $attn_{rot} = 7$, respectively, for the feature maps of translation and rotation. For the transformer module, we configured 6 encoder layers and 4 decoder layers. The hidden layer of the regression head for the translation branch is set to 256, while the rotation branch is assigned the value of 512. During training, the decoder's output is selected based on the index of the real scene, while the estimated scene index that only used for computing the NNL. However, when performing inference, the scene index is unknown. We select the index with the highest probability predicted by the model. We have conducted all of our experiments utilizing a single Nvidia 3080Ti GPU.

B. DATASET

We implemented a comprehensive evaluation of our model's performance, leveraging the 7Scenes [26] and Cambridge Landmarks [8] datasets as our benchmarks. The 7Scenes dataset encompasses seven distinct, small-scale indoor environments. It utilizes Kinect sensors to capture multiple sequences of 500 to 1000 frames within the room. This dataset contains images with repetitive and texture-less regions, which can pose challenges for APR algorithms. The Cambridge Landmarks dataset comprises a collection of six large-scale outdoor scenes captured, all captured using a mobile phone. The various factors in the images pose difficulties for APR algorithms. We selected four benchmark scenes for comparison, which are typically considered as typical evaluation scenes for APR.

C. COMPARISON WITH EXISTING METHODS

Our method aims to enhance the accuracy of APR techniques by considering the interaction between local and global attention in images. Therefore, we compared our approach with SOTA multi-scene APRs on the indoor and outdoor benchmark datasets. We utilized the median error of each individual scene as a measure of prediction accuracy, which reflects the central tendency of the data. Position error

TABLE 2. Evaluation of localization accuracy in comparison to other typical multi-scene APRs on Cambridge landmark dataset, represented as median error ($m/\circ$). Best results are highlighted in bold.

Method	King's College	Old Hospital	Shop Facade	St.Mary
MSPN [15]	1.73/3.65	2.55/4.05	2.92/7.49	2.67/6.18
MS-Tran [16]	0.83/1.47	**1.81**/2.39	0.86/**3.07**	1.62/3.99
Auto-Encoder [17]	0.90/1.49	2.07/2.58	0.99/3.88	1.64/4.16
H-AttnAPR(ours)	**0.78/1.42**	1.83/**2.12**	**0.82**/3.20	**1.48/3.91**

TABLE 3. Evaluation of localization accuracy in comparison to other typical multi-scene APRs on 7Scenes dataset, represented as median error ($m/\circ$). Best results are highlighted in bold.

Method	Chess	Fire	Heads	Office	Pumpkin	Kitchen	Stairs
MSPN [15]	**0.09**/4.76	0.29/10.50	0.16/13.10	**0.16**/6.80	0.19/5.50	0.21/6.61	0.31/11.63
MS-Trans [16]	0.11/4.66	0.24/9.60	0.14/12.19	0.17/5.66	0.18/4.44	**0.17/5.69**	0.26/8.54
Auto-Encoder [17]	0.12/4.95	0.24/9.31	0.14/12.50	0.19/5.79	0.18/4.48	0.18/6.19	0.25/8.74
H-AttnAPR(ours)	**0.09/4.47**	**0.22/9.20**	**0.12/11.80**	**0.16/5.60**	**0.16/4.40**	**0.17**/5.88	**0.23/7.68**

TABLE 4. Evaluation of localization accuracy in comparison to other typical APRs(single-scene and multi-scene) on Cambridge Landmark dataset, represented as average median error ($m/\circ$). Best results are highlighted in bold.

Method	Average	Ranks
Single-scene APRs		
PoseNet [8]	2.09/6.84	7/8
MapNet [27]	1.63/3.64	6/5
LSTM [10]	1.30/5.52	3/7
IRPNet [11]	1.42/3.45	5/4
Multi-scene APRs		
MSPN [15]	2.47/5.34	8/6
MS-Tran [16]	1.28/2.73	2/2
Auto-Encoder [17]	1.40/3.01	4/3
H-AttnAPR(ours)	**1.23/2.66**	**1/1**

TABLE 5. Evaluation of localization accuracy in comparison to other typical APRs(single-scene and multi-scene) on 7Scenes dataset, represented as average median error ($m/\circ$). Best results are highlighted in bold.

Method	Average	Ranks
Single-scene APRs		
PoseNet15 [8]	0.44/10.40	11/11
MapNet [27]	0.21/7.77	6/5
LSTM [10]	0.31/9.85	9/10
MLFBPPose [28]	0.22/9.29	8/9
AtLoc [13]	0.20/7.56	4/4
IRPNet [11]	0.32/8.49	10/8
NeuralR-Pose [29]	0.21/7.92	6/6
Multi-scene APRs		
MSPN [15]	0.20/8.41	4/7
MS-Trans [16]	0.18/7.28	2/2
Auto-Encoder [17]	0.19/7.42	3/3
H-AttnAPR(ours)	**0.16/7.00**	**1/1**

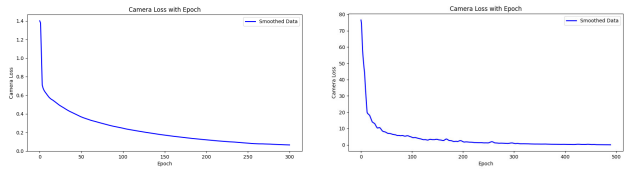
and orientation error are the Euclidean distances between the predicted position and the actual position, respectively. Additionally, we also compared our method with classical single-scene APR approaches to illustrate the effectiveness of our method. Since directly predicting poses from a single image has significantly lower localization accuracy compared to methods based on feature matching [7] or those leveraging additional data during training and inference [3], we only compared with APR methods that rely solely on a single image for inference.

Table 2 and Table 3 showcase the performance of our method compared to advanced multi-scene methods on the Cambridge Landmarks and 7Scenes datasets, respectively. We report the median position error and orientation error for single scene. The data presented in the tables indicate that our model surpasses current multi-scenario pose estimation approaches with reduced position and orientation errors, consistently achieving greater positioning accuracy. We believe that our model has an advantage in feature extraction. Table 4 and Table 5 show the comparison between our method and the other known classic and advanced APRs for single- and multi-scene. We use average position and orientation errors and rankings of all scenes in indoor and outdoor datasets to demonstrate the effectiveness of the methods. Using PoseNet's predicted position and orientation error results as the baseline, our algorithm has achieved a 41.1% reduction in translation error and a 61.1% decrease in rotation error. The tables show that our method is ahead of other methods.

D. TRAINING CURVE AND ATTENTION MAP VISUALIZATION

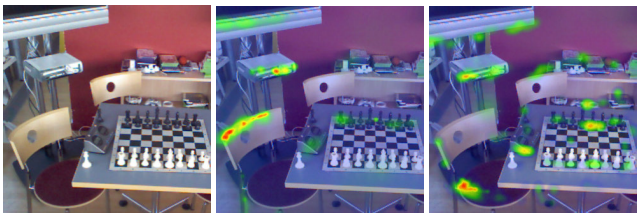
The training curve of our model is shown in Figure 4. As can be seen from the figure, the loss value is relatively high at the beginning due to initialization, but it gradually decreases and stabilizes with the increase of training epochs. To intuitively

understand the reason behind the improvements, we visualize the attention maps of an image taken from the chess scene. As shown in Fig.5, benefiting from the HA model, our method focuses on regions with structural and textural information on the imageas well as the local patches within the image.



(a) 7Scenes (b) Cambridge Landmark

FIGURE 4. Training curve of our model.



(a) Input image (b) Position (c) Rotation

FIGURE 5. Visualization of attention heat map.

E. ABLATION STUDY

In this section, we primarily delve into the HA module and the impact of the grid size in the first layer of HA on the model, as this module is a key component of our model. Additionally, we conducted ablation experiments on the model's batch size and the influence of hierarchical attention under different CNN backbone networks on model performance.

1) THE MODULE

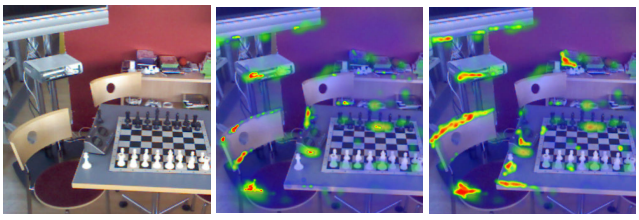
To demonstrate the effectiveness of the HA module, we have carried out an ablation study on the 7Scenes dataset, and the results are presented in Table6. We reported the average median error across all scenes. In (a), without adding any additional operations, the features extracted directly from the CNN backbone were fed into the scene classification module. In contrast, in (b), we employed the HA module, which led to a reduction in both translation and rotation errors, achieving better performance. To vividly illustrate the module's effectiveness, we have visualized the activation maps after model processing, that is, before the step (f.) in Figure 2, using heatmaps. Figure 6(b) shows the model without the HA module, while Figure 6(c) displays the model with the HA module integrated. We superimpose attention heatmap representations of translation and rotation to highlight how the model represents key information in the image. By comparing these two figures, it is evident that with the addition of the HA module, the model is able to capture more key informational cues from the images.

TABLE 6. Ablations of whether to use HA module validated on the 7Scenes dataset. We present the average median errors in both position and orientation, encompassing all tested scenes. The selected model is emphasized in bold. (*m*/∘).

HA Model	Position(*m*)	Orientation(∘)
a	0.19	7.37
b	**0.16**	**7.00**

TABLE 7. Ablations of the grid-size of HA model validated on the 7Scenes dataset. We present the average median errors in both position and orientation, encompassing all tested scenes. The selected choice is highlighted in bold. (*m*/∘).

grid-size t/rot	Position(*m*)	Orientation(∘)
2/2	0.17	7.90
2/4	**0.16**	**7.00**
7/7	0.18	7.63



(a) Input image (b) without HA (c) with HA

FIGURE 6. Visualizing the HA module represents image information in the model.

2) GRID SIZE SETTING

In studying the performance of our HA module, we conducted an ablation study on the performance of different grid size values for indoor scenes. The results are presented in Table 7. We observed that the best performance is achieved when set the grid-size $= 2$ for the translation branch and grid-size $= 4$ for the rotation branch. We believe that in indoor scenes, camera translation typically involves shorter distances, while rotation may cover a larger range of angles. Due to the relatively enclosed and limited space in indoor environments, the camera may only need to capture changes in local details during translation, whereas during rotation, a broader perspective and the continuity of the scene need to be considered. Therefore, using different grid sizes for translation and rotation in indoor environments can more effectively capture subtle changes in the scene, thereby enhancing the model's understanding of indoor environments. This differentiated approach helps the model better adapt to the complexity and diversity of indoor scenes.

3) BATCH SIZE

Generally speaking, the amount of data fed into the model during each training iteration can have a significant impact

TABLE 8. Ablations of batch size Validated on the 7Scenes dataset. We present the average median errors in both position and orientation, encompassing all tested scenes. The selected choice is highlighted in bold. (*m*/◦).

Batch Size	Position(*m*)	Orientation(◦)
6	0.18	7.48
8	0.16	7.28
12	**0.16**	**7.00**
16	0.17	7.60

TABLE 9. Ablations of CNN backbone validated on the 7Scenes dataset. We present the average median errors in both position and orientation, encompassing all tested scenes. The selected choice is highlighted in bold. (*m*/◦).

CNN Backbone	Position(*m*)	Orientation(◦)
Resnet50	0.17	7.58
EffcientNetB0	**0.16**	**7.00**
EffcientNetB1	0.15	6.96

on the training effect of the model. To verify the influence of different batch sizes on our experiments, we conducted an ablation study on batch size, and experimental results are shown in Table 8. We set the batch size to 6, 8, 12, and 16. As can be seen from the table, the model's performance does not always improve with the increase in batch size; instead, it reaches the best performance when the batch size is set to 12. We believe that setting the batch size too large may cause the model to prematurely converge to a local minimum, while setting it too small may lead to overfitting during training, both of which can make the training process unstable.

4) CONVOLUTION BACKBONE

We have validated our model using three convolutional networks: ResNet50, EfficientNetB0, and EfficientNetB1. The results obtained from training different CNNs are shown in Table 9. Compared to ResNet50, the EfficientNet networks demonstrate higher performance, which could be due to overfitting (with 26M parameters for ResNet50, EfficientNetB0 and EfficientNetB1 have 5.3M and 7.8M parameters [30], respectively) or a better learning capability. The best performance was achieved using EfficientNetB1, indicating that better performance can be obtained by appropriately increasing the depth of the network, but this comes at the cost of corresponding memory and time.

F. ANALYSIS OF NOISE RESISTANCE AND EXECUTION TIME

1) NOISE RESISTANCE

To verify the noise resistance capability of our model, we added Gaussian noise of varying intensities to our test set. We then validated the noisy images using our model. Figure 7 shows a comparison of images with different noise intensities, and Table 10 presents the results of our model's performance on the 7Scenes test set with added noise, taking the average error across seven scenes. As can be seen from the table, our model's performance does not significantly change when the Gaussian noise intensity is set to 5 and 10. However, when the Gaussian noise is set to 15, there is a decline in performance, but it still achieves advanced results in the field of multi-scene camera absolute pose estimation models. We consider that the addition of Gaussian noise has obscured the true information in the images, thereby blurring the image features and causing a slight decline in performance. To enhance the robustness of our model, we added noise of varying intensities during the training process. The experimental results demonstrate that our model has good adaptability to noise.

TABLE 10. Comparison study on gaussian noise We present the average median errors in both position and orientation, encompassing all tested scenes. (*m*/◦).

Gaussian noise intensity	Position(*m*)	Orientation(◦)
5	0.16	7.00
10	0.16	7.15
15	0.18	7.28

(a) $\sigma = 0$ (b) $\sigma = 5$ (c) $\sigma = 10$ (d) $\sigma = 15$

FIGURE 7. Visualization of adding gaussian noise to an image.

TABLE 11. The results of the model's runtime (ms) and memory(Mb) as the number of learned scenes varies. In this instantiation, we used hierarchical attention, a six-layer encoder, and a four-layer decoder.

Number of scenes	Runtime(*ms*)	Memory(*Mb*)
1	36.4	74.8
4	36.6	74.8
7	36.7	74.8
11	36.7	74.8
100	36.8	75.0

2) EXECUTION TIME

Multi-scene APRs have two main advantages over other localization methods: runtime and memory footprint. Multi-scene APRs only require deployment of a single model for both training and inference, whereas single-scene APRs need N models, such as the PoseNet model [8], which would require approximately 100 × 50Mb, or about 5Gb, of storage space to serve a location with 100 scenes. To evaluate the scalability of our model, we measured its runtime and memory usage as the number of scenes increased. For this, we instantiated models for an increasing number of scenes and measured their memory consumption and forward propagation runtime, allowing us to assess the scalability of our method without real data. The results of this experiment are shown in Table11. The memory usage of our model remained relatively constant, increasing by only 0.2Mb when going from 4 to 100 scenes. The model's runtime remained

constant in the range of 4-100 scenes, which was as expected, as the hierarchical attention module's processing of images in blocks greatly reduced computational complexity.

V. CONCLUSION

In this paper, we present H-AttnAPR, a modified APR model that leverages a HA mechanism to enhance the precision of image-based positioning and orientation prediction, critical for various industrial applications such as robotic navigation, autonomous vehicle systems, and precision manufacturing. The HA module integrates multi-level features to create robust and discriminative image representations, which is essential for achieving high-accuracy localization in complex industrial settings where traditional APRs may fall short. Our experiments across various indoor and outdoor datasets demonstrate that H-AttnAPR achieves superior positioning accuracy compared to SOTA APRs. This illustrates its potential to significantly enhance industrial applications, such as refining the precision of robotic arms and enabling seamless autonomous navigation in logistics. Future work will investigate integrating relative motion between consecutive image frames with relative camera pose estimation to further enhance localization accuracy and meet the dynamic and complex demands of industrial environments.

ACKNOWLEDGMENT

The authors acknowledge these funding sources for their support.

REFERENCES

[1] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. ECCV*. Cham, Switzerland: Springer, Jan. 2020, pp. 213–229.

[2] E. Brachmann and C. Rother, "Visual camera re-localization from RGB and RGB-D images using DSAC," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 5847–5865, Apr. 2021.

[3] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From coarse to fine: Robust hierarchical localization at large scale," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12708–12717.

[4] W. Gao, F. Wan, X. Pan, Z. Peng, Q. Tian, Z. Han, B. Zhou, and Q. Ye, "TS-CAM: Token semantic coupled attention map for weakly supervised object localization," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 2866–2875.

[5] A. Nasrinahar and J. H. Chuah, "Intelligent motion planning of a mobile robot with dynamic obstacle avoidance," *J. Vehicle Routing Algorithms*, vol. 1, nos. 2–4, pp. 89–104, Nov. 2018.

[6] H. O. Khogali and S. Mekid, "The blended future of automation and AI: Examining some long-term societal and ethical impact features," *Technol. Soc.*, vol. 73, May 2023, Art. no. 102232.

[7] E. Brachmann, A. Krull, S. Nowozin, J. Shotton, F. Michel, S. Gumhold, and C. Rother, "DSAC—Differentiable RANSAC for camera localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2492–2500.

[8] A. Kendall, M. Grimes, and R. Cipolla, "PoseNet: A convolutional network for real-time 6-DOF camera relocalization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2938–2946.

[9] Y. Shavit, R. Ferens, and Y. Keller, "Coarse-to-fine multi-scene pose regression with transformers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 12, pp. 14222–14233, Dec. 2023.

[10] F. Walch, C. Hazirbas, L. Leal-Taixé, T. Sattler, S. Hilsenbeck, and D. Cremers, "Image-based localization using LSTMs for structured feature correlation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 627–637.

[11] Y. Shavit and R. Ferens, "Do we really need scene-specific pose encoders?" in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 3186–3192.

[12] A. Kendall and R. Cipolla, "Geometric loss functions for camera pose regression with deep learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6555–6564.

[13] B. Wang, C. Chen, C. X. Lu, P. Zhao, N. Trigoni, and A. Markham, "AtLoc: Attention guided camera localization," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, Apr. 2020, pp. 10393–10401.

[14] M. Cai, C. Shen, and I. Reid, "A hybrid probabilistic model for camera relocalization," in *Proc. BMVC*, 2018.

[15] H. Blanton, C. Greenwell, S. Workman, and N. Jacobs, "Extending absolute pose regression to multiple scenes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 170–178.

[16] Y. Shavit, R. Ferens, and Y. Keller, "Learning multi-scene absolute pose regression with transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 2713–2722.

[17] Y. Shavit and Y. Keller, "Camera pose auto-encoders for improving pose regression," in *Proc. ECCV*. Springer, Jan. 2022, pp. 140–157.

[18] Y. Liu, Y.-H. Wu, G. Sun, L. Zhang, A. Chhatkuli, and L. Van Gool, "Vision transformers with hierarchical attention," *Mach. Intell. Res.*, vol. 21, no. 4, pp. 670–683, Aug. 2024.

[19] S. El Hazzat and M. Merras, "Improvement of 3D reconstruction based on a new 3D point cloud filtering algorithm," *Signal, Image Video Process.*, vol. 17, no. 5, pp. 2573–2582, Jul. 2023.

[20] V. Balntas, S. Li, and V. A. Prisacariu, "RelocNet: Continuous metric learning relocalisation using neural nets," in *Proc. ECCV*, Jan. 2018, pp. 782–799.

[21] Z. Laskar, I. Melekhov, S. Kalia, and J. Kannala, "Camera relocalization by computing pairwise relative poses using convolutional neural network," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2017, pp. 920–929.

[22] M. Ding, Z. Wang, J. Sun, J. Shi, and P. Luo, "CamNet: Coarse-to-fine retrieval for camera re-localization," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 2871–2880.

[23] E. Brachmann and C. Rother, "Expert sample consensus applied to camera re-localization," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 7524–7533.

[24] H. Fan, B. Xiong, K. Mangalam, Y. Li, Z. Yan, J. Malik, and C. Feichtenhofer, "Multiscale vision transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 6804–6815.

[25] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9992–10002.

[26] B. Glocker, S. Izadi, J. Shotton, and A. Criminisi, "Real-time RGB-D camera relocalization," in *Proc. IEEE Int. Sympos. Mix. Augment. Reality (ISMAR)*, Dec. 2013, pp. 173–179.

[27] S. Brahmbhatt, J. Gu, K. Kim, J. Hays, and J. Kautz, "Geometry-aware learning of maps for camera localization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2616–2625.

[28] X. Wang, X. Wang, C. Wang, X. Bai, J. Wu, and E. R. Hancock, "Discriminative features matter: Multi-layer bilinear pooling for camera localization," in *Proc. BMVC*, Jul. 2019, pp. 1–13.

[29] Y. Zhu, R. Gao, S. Huang, S.-C. Zhu, and Y. N. Wu, "Learning neural representation of camera pose with matrix representation of pose shift via view synthesis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 9954–9963.

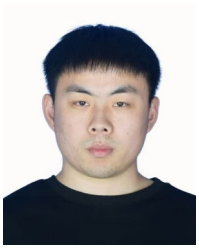
[30] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. ICML*, Jan. 2019, pp. 6105–6114.

XINHUA LU received the B.E., M.E., and Ph.D. degrees from Zhengzhou University, Zhengzhou, China, in 2003, 2007, and 2019, respectively. Since 2007, he has been working with Nanyang Institute of Technology, Nanyang, China. He is currently working as an Associate Professor with Electronic Information Institute. His research interests include wireless communication, SLAM, and machine learning.

JINGUI MIAO received the B.S. degree from Cyberspace Security Academy, Zhengzhou University, Zhengzhou, China, in 2022, where she is currently pursuing the M.S. degree with the School of Computer and Artificial Intelligence. Her research interests include artificial intelligence and camera pose estimation.

QINGJI XUE received the Ph.D. degree from the School of Computer Science and Technology, Wuhan University of Technology, Wuhan, China, in 2011. He is currently working with the School of Digital Media and Art Design, Nanyang Institute of Technology. His current research interests include computer vision and the IoT technology.

HUI WAN received the B.S. degree from International College, Zhengzhou University, Zhengzhou, China, in 2022, where he is currently pursuing the M.S. degree with the School of Computer and Artificial Intelligence. His research interests include artificial intelligence and 3D point cloud registration.



HAO ZHANG received the B.S. degree from the School of Information Engineering, Tianjin University of Commerce. He is currently pursuing the master's degree with the School of Computer and Artificial Intelligence and Software, Zhengzhou University. His research interest includes computer vision.