Pattern Recognition 146 (2024) 109975

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

TransBoNet: Learning camera localization with Transformer Bottleneck and Attention

Xiaogang Song [a], Hongjuan Li [a], Li Liang [a], Weiwei Shi [a], Guo Xie [b], Xiaofeng Lu [a], Xinhong Hei [a,*]

[a] School of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048, China
[b] School of Automation Information and Engineering, Xi'an University of Technology, Xi'an 710048, China

ARTICLE INFO

Keywords:
Camera localization
6DoF pose
Hybrid attention
Pose regression

ABSTRACT

6DoF camera localization is an important component of autonomous driving and navigation. Deep learning has achieved impressive results in localization, but its robustness in dynamic environments has not been adequately addressed. In this paper, we propose a framework based on hybrid attention mechanism which can be generally applied to existing CNN-based pose regressors to improve their robustness in dynamic environments. Specifically, we propose a novel Transformer Bottleneck (TBo) block including convolution, channel attention, and a position-aware self-attention mechanism, which extracts more geometrically robust features by capturing the corresponding long-term dependencies between pixels. Furthermore, we introduce shuffle attention (SA) before the pose regressor, which integrates feature information in both spatial and channel dimensions, forcing the network to learn geometrically robust features, reducing the effects of dynamic objects and illumination conditions to improve camera localization accuracy. We evaluate our method on commonly benchmarked indoor and outdoor datasets and the experimental results show that our proposed method can significantly improve localization performance compared compare favorably to contemporary pose regressors schemes. In addition, extensive ablation evaluations are conducted to prove the effectiveness of our proposed hybrid attention bottleneck block for pose regression networks.

1. Introduction

Estimating the pose of the camera from the captured image is one of the essential tasks in many computer vision applications. Accurate estimation of the absolute pose of the camera is the key to the application of augmented reality, autonomous driving [1,2] and indoor navigation, in which location information is of key important to its performance.

Traditional algorithms for camera pose estimation are solved by using 3D structure-based methods, indirect-direct methods [3,4] or image retrieval [5]. The method based on 3D structure first establishes correspondences between pixels in the test image and 3D points in the scene. Then, these 2D-3D matches are used to estimate the camera pose by applying the Perspective-n-Point (PnP) solver and a RANSAC algorithms [6]. Image retrieval is first applied to extract candidate images from a large database of geo referenced images. Local features are fetched from query and geo referenced images and then matched [7]. However, when deployed in outdoor scenes, the performance will degrade. This is due to the traditional geometric localization methods mainly use handcrafted features and descriptors, so they are rather sensitive to viewpoint change, illumination variation, and dynamic objects, which are apparent in unconstrained environments. Recently, Deep neural networks (DNNs) based camera positioning method have drawn increasing attention because of its low cost and broad applicability, which also urges researchers to exploit the potential of DNNs in visual localization. There are more and more methods based on deep learning, which can automatically extract image features and directly regress absolute camera pose from a single image. As shown in PoseNet [8], features are generated on a single image, and finally global pose is output, that is, pose prediction is regarded as a regression problem, and its variant models use different feature extraction networks [9], or geometric constraints [10]. Although these technologies generally show good performance, they lack robustness in the face of dynamic objects or lighting changes, especially in outdoor datasets with highly variable scenes. In order to solve the problem of insufficient robustness, further technologies have considered using multiple images as the input of the network, on the premise that the network can learn to reject the features of time inconsistency across frames. There are also similar approaches that use time-context that is inherently given by consecutive images. For example, VidLoc [11], MapNet [12] and PoseNet + LSTM [13] have achieved good performance in camera pose regression.

* Corresponding author.
E-mail address: heixinhong@xaut.edu.cn (X. Hei).

https://doi.org/10.1016/j.patcog.2023.109975
Received 24 December 2021; Received in revised form 10 September 2023; Accepted 12 September 2023
Available online 21 September 2023

In this work, we pursue another approach to realize camera pose regression, which can achieve or even exceed the performance of multi-frame continuous technology by learning to focus on time consistent and information rich image parts (such as buildings). We propose a new attention-based neural network, which uses a single image as network input and ignores the dynamic parts such as vehicles and pedestrians in feature extraction. Different from previous methods, our method does not need to manually design multiple frames or geometric constraints.

In this paper, we show that our model is superior to the previous technology. It can work effectively in both indoor and outdoor scenes, and is simple and end-to-end training without any handmade geometric loss functions. We provide detailed network structure information to understand how to integrate attention into the network, so as to achieve accurate and robust camera pose estimation.

The main contributions of this work are three folds: (1) We propose a novel Transformer Bottleneck block with self-attention and channel attention to overcome the limitations of convolution. This coupling allows them to be optimized in a mutually reinforcing manner, significantly improving fine-grained feature extraction for accurate localization. (2) We propose a novel end-to-end hybrid attention network for single image localization, which improves the accuracy and robustness of camera localization, especially in dynamic scenes. (3) We conduct extensive experiments in both indoor and outdoor dataset, which shows that our model performs better than the existing competitive methods.

The rest of this paper is organized as follows: Section 2 reviews some related works in camera localization. Section 3 describes the basic idea of attention-based camera localization method and its loss function items in detail. We introduce our experiments and compare them with many current methods in Section 4. Finally, we concluding remarks and prospects for future work in Section 5.

2. Related work

2.1. Deep neural networks for camera localization

The method of deriving Degree of Freedom (6DoF) pose directly from image has been studied for decades. Therefore, there are many classical methods whose complex components are replaced by machine learning or deep learning. Compared with traditional structure-based methods [14] and image retrieval based global descriptors methods [15,16], Deep Neural Networks (DNNs) can automatically learn features from data without manually building a map or landmark feature database [17,18]. In [8], PoseNet is introduced, which is an end-to-end 6-dof monocular camera pose estimation network. The proposed algorithm maps monocular images firstly to a high-dimensional space, which is linear in camera pose and robust to annoyance variables. After that, the 6DoF camera pose can be regressed from this representation without any hand-engineering effort. However, its accuracy is far lower than that of 3D based methods. In order to improve the positioning accuracy, contemporary absolute pose estimation has studied different Convolution Neural Network (CNN). Then, the method is extended to spatial [19] and temporal [11] by inverse RNN (such as LSTM). PoseNet + LSTM [13] uses LSTM units on the CNN output, which shows the utility of structured dimensionality reduction of feature vectors, it helps to improve the camera localization performance to a great extent. Subsequently, by using Bayesian CNN [20,21] to estimate the attitude uncertainty in the global camera, and replacing the feature extraction architecture with residual neural network, the positioning performance is further improved [9]. However, the above method relies on manually adjusted scale factors to balance the position and rotation loss in the learning process. In order to solve this problem, learning weighted loss and geometric re projection loss. To solve this problem, learn weighted loss and geometric reprojection loss [10] to produce more accurate results. Recent work has also used the geometric constraints of paired images [22,23], increased training data through synthetic generation, or introduced neural maps Pose graph optimization of the model [24]. We develop a DNN-based automatic adjustment mechanism for camera localization and automatically learn geometrically robust features that facilitate pose regression, instead of imposing temporal information or geometric constraints as in previous work. Our model outperforms the existing methods in pose regression by a large margin.

2.2. Attention and transformer in image analysis

The importance of attention has been studied extensively in the previous literature. It favors the assignment of the most informative feature expressions while suppressing the less useful ones. Self-attention methods calculate the context of a position as a weighted sum of all positions in the image. SENet [25] provides an effective, lightweight gating mechanism to self-recalibrate feature maps via channel-wise importance. ECA-Net [26] adopts a 1-D convolution filter to generate channel weights, which significantly reduces the model complexity of Squeeze-and-Excitation (SE). Wang et al. [27] proposed a non-local (NL) module to generate a large number of attention maps by calculating the correlation matrix between each spatial point in the feature map. GCNet [28] fully explores the advantages and disadvantages of the non-local and SE modules, and combines the advantages of both to design a more effective global context module, which achieves convincing results on object detection tasks. CBAM [29], SANet [30], STAL [31] and SGE [32] combine spatial attention and channel attention serially, while DANet [33] adaptively integrates local features with their global dependencies by summing two attention modules from different branches. Our work introduces multi-head self-attention mechanism and channel attention mechanism into the camera localization model to show the effectiveness of correlating robust key features and improve model performance.

3. Camera localization with transformer bottleneck and attention module

This section provides an overview of the proposed framework explaining its most distinctive features. A deep network architecture based on hybrid attention, which is used to learn camera poses from a single image. Fig. 1 illustrates a pipeline of the proposed framework, consisting of a feature encoder, an attention module and a pose regressor. First, the Transformer Bottleneck in the feature encoder can extract the global features in a given single image. This can be attributed to the fact that attention not only considers the content information, but also the relative distances between features at different locations, which can effectively correlate the information across objects with location awareness. Second, shuffle attention module [30] constructed by attention mechanism are used to stand out the most important features from different channels to highlight regions that are greatly relevant to camera localization. Finally, the pose regressor further maps the aggregation features after the attention operator to the camera pose, i.e. the 3-D location and 4-D quaternion (orientation). The final networks are supervised by the distance between the ground truth and the predicted poses.

3.1. Feature encoder

The Feature Encoder is used to extract the features required for pose regression task from a single monocular image. Previous works shows the successful application of classical convolutional neural network (CNN) architectures in camera pose regression. Specifically, in order to find a balance between the number of network parameters and accuracy, we use ResNet50 architecture as our basic network, which achieves more stable and accurate localization results than other structures, due to the residual network can train the deep level of the neural network and reduce the problem of gradient vanishing. Therefore, we consider using ResNet50 as the basic network of encoder in the

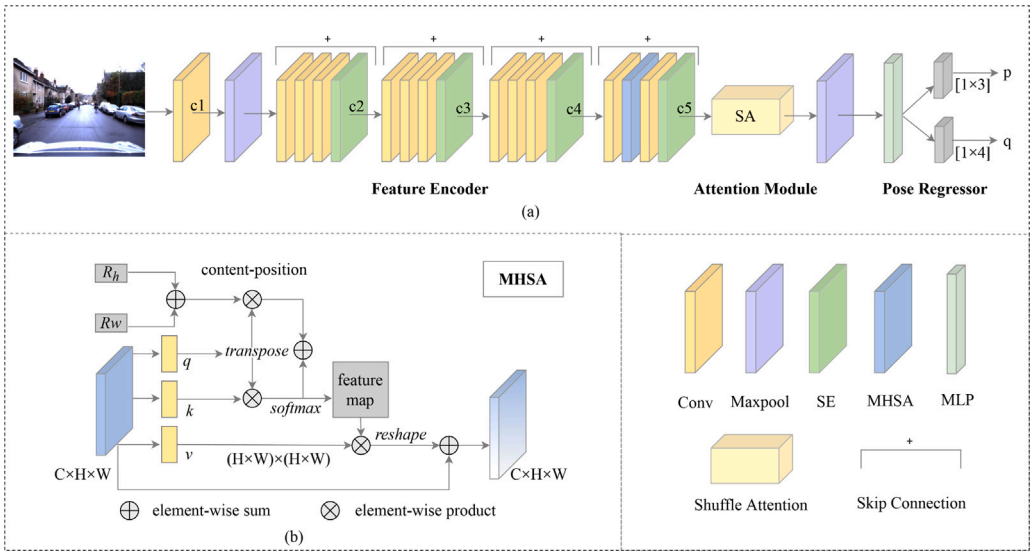


Fig. 1. (a) An overview of our proposed TransBoNet framework, consisting of Feature Encoder (extracts local features from a single image), Attention Module (computes the attention and reweights the features), and Pose Regressor (maps the new features into the camera pose, position p or orientation q). (b) Structure of the Position-aware MHSA.

Table 1
Comparison of structural parameters of Feature Encoder.

Stage	Output	ResNet50	Feature Encoder
c1	512 × 512	7 × 7, 64, stride 2	7 × 7, 64, stride 2
c2	256 × 256	$\begin{Bmatrix} 1\times1,\ 64 \\ 3\times3,\ 64 \\ 1\times1,\ 256 \end{Bmatrix}\times3$	$\begin{Bmatrix} 1\times1,\ 64 \\ 3\times3,\ 64 \\ 1\times1,\ 256 \\ SE,\ 256 \end{Bmatrix}\times3$
c3	128 × 128	$\begin{Bmatrix} 1\times1,\ 128 \\ 3\times3,\ 128 \\ 1\times1,\ 512 \end{Bmatrix}\times4$	$\begin{Bmatrix} 1\times1,\ 128 \\ 3\times3,\ 128 \\ 1\times1,\ 512 \\ SE,\ 512 \end{Bmatrix}\times4$
c4	64 × 64	$\begin{Bmatrix} 1\times1,\ 256 \\ 3\times3,\ 256 \\ 1\times1,\ 1024 \end{Bmatrix}\times6$	$\begin{Bmatrix} 1\times1,\ 256 \\ 3\times3,\ 256 \\ 1\times1,\ 1024 \\ SE,\ 1024 \end{Bmatrix}\times6$
c5	32 × 32	$\begin{Bmatrix} 1\times1,\ 512 \\ 3\times3,\ 512 \\ 1\times1,\ 2048 \end{Bmatrix}\times3$	$\begin{Bmatrix} 1\times1,\ 512 \\ MHSA,\ 512 \\ 1\times1,\ 2048 \\ SE,\ 2048 \end{Bmatrix}\times3$

proposed model. To encourage the learning of meaningful features for pose regression, we modify the base network. First, we embed a SE layer in each residual block, and then replace the 3 × 3 spatial convolution in the c5 stage with a Position-aware Multi-Head Self-Attention (MHSA) layer. Finally, the base network is further modified by replacing the final 1000 dimensional fully-connected layer with a 2048 dimensional fully-connected layer and removing the Softmax layers used for classification. The specific structural parameters are shown in Table 1. Given an image $I \in R^{(H\times W\times C)}$, the features of $X \in R^{(8\times8\times2048)}$ can be extracted by the Feature Encoder.

3.1.1. Transformer bottleneck block

Previous methods [8,21] only exploit high-level features to regress camera pose, which will result in low localization accuracy due to ambiguous patterns caused by visual similarity. We argue that a robust feature representation containing abundant structure information and spatial details can address such issue.

Through the Feature Encoder, we can obtain different level feature maps with different receptive fields. The low-level feature maps with abundant structure information can deal with the ambiguities, while high-level feature maps with rich semantic information can be used to model the pose regression problem accurately. Therefore, we argue that effectively fusing the structure and semantic information extracted from different receptive fields is important for improving the accuracy of camera localization. However, since the different feature representations of low-level and high-level feature maps, it is not easy for the network to learn and generalize using simple pixel-to-pixel addition to fuse them. In contrast, we propose a novel Transformer Bottleneck block embedded with a position-aware multi-head self-attention module and a channel attention module to learn a strong and distinctive feature representation, through which we can bridge gap between structure details and semantic information. We compare the structure of the two bottleneck blocks, as shown in Fig. 2.

Channel Attention Module: It uses scalars to represent and evaluate the importance of each channel, enabling the model to filter out worthless feature channels. Through capturing the feature dependencies in channel dimension, the redundant information is largely reduced and the most important information of the input image is retained. Moreover, inspired by the work of Hu et al. [34], we embed SE layers into each residual block to learn a strong and distinctive feature representation, as shown in Fig. 2(b), a structure that retains shallow layers information and passes it to deep layers. In addition, it can not only fuse all features but also adaptively learn different weights of different level feature information.

Position-aware Multi-Head Self-Attention Module: Previous work [35] considered the input resolution (224 × 224(for classification) and 640 × 640 (for detection experiments in SASA)), the method of using self-attention in the whole backbone is feasible. Therefore, we consider the use of self-attention for visual localization tasks as an alternative to convolutions. Considering that $O(n^2d)$ memory and computation are required to perform self-attention globally in n entities [36], we believe that the simplest setting to comply with the above factors is to incorporate self-attention into the trunk with the lowest resolution feature mapping, that is, the three bottleneck blocks in c5 stack. Replacing spatial 3 × 3 convolution in the bottleneck block with MHSA layers constitutes the basis of the Feature Encoder, as shown in Fig. 2(b).

3.1.2. Position-aware multi-head self-attention

Given an input tensor of shape (H, W, d_{in}), we flatten it to a matrix $X \in R^{HW\times d_{in}}$ and perform MHSA as proposed in the Transformer

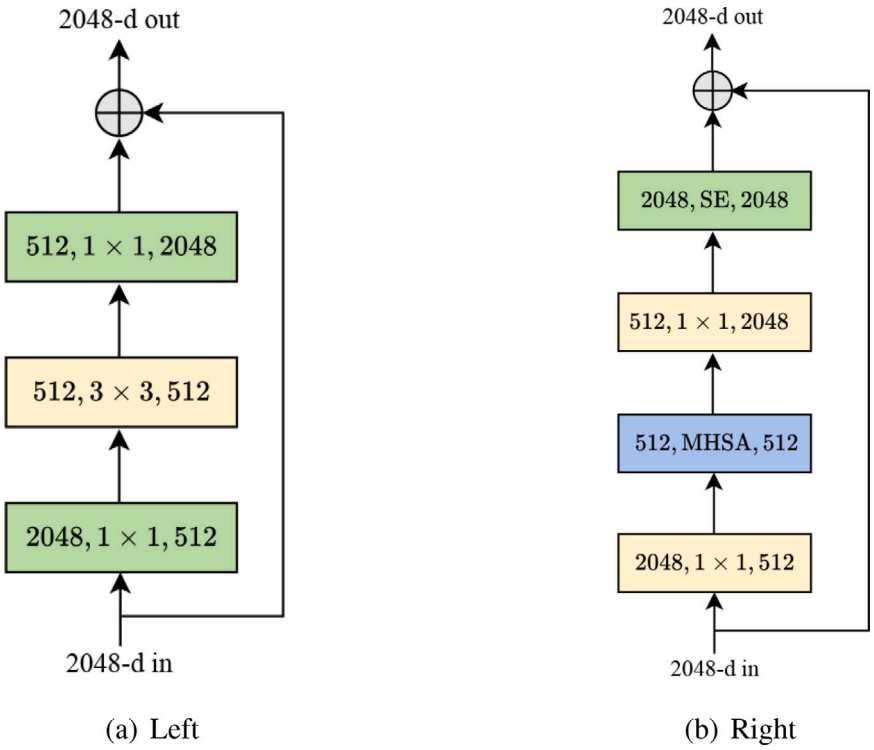


Fig. 2. Left: A ResNet Bottleneck Block, Right: A Transformer Bottleneck block.

architecture [36]. d_{in} refers to the number of input filters of an activation map. Single-headed attention mechanism for computing the pixel output is then computed as follows (see Fig. 1(b)):

$$O_h = Softmax\left(\frac{QK^T}{\sqrt{d_k^h}}\right)V \tag{1}$$

where the queries $Q = XW_q$, keys $K = XW_k$ and values $V = XW_v$ are linear transformations of the pixel in position ij and the neighborhood pixels. $W_q, W_k \in R^{d_{in}\times d_k^h}$ and $W_v \in R^{d_{in}\times d_v^h}$ are parameter matrices. These parameter matrices are unique per layer and attention head. The outputs of all heads are then concatenated and projected again as follow:

$$MHSA(X) = Concat(O_1, \ldots, O_h)W^O \tag{2}$$

Where $W^O \in R^{hd_v\times d_v}$ is parameter matrix. In this work we employ $h = 4$ parallel attention, or heads. $MHSA(X)$ is then reshaped into a tensor of shape (H, W, d_v) to match the original spatial dimensions.

To make the attention operation position aware, the Transformer based architecture usually adopts position encoding. Recently, it has been considered that relative position encodings [37] are more suitable for visual tasks. This can be attributed to the fact that attention considers not only the content information, but also the relative distance between different location features. Therefore, it can effectively connect information across object with positional awareness. In the MHSA module, we apply the 2D relative position self-attention in [35]. We implement two-dimensional relative self-attention by independently adding relative height information and relative width information. The output of head h is now defined as

$$O_h = Softmax\left(\frac{QK^T + R_w + R_h}{\sqrt{d_k^h}}\right)V \tag{3}$$

Where $R_w, R_h \in R^{HW\times HW}$ are matrices of relative position logits along width and height dimensions that satisfy $R_w[i,j] = q_i^T r^W_{j_a - i_a}$ and $R_h[i,j] = q_i^T r^W_{j_b - i_b}$. $r^W_{(j_a - i_a)}$ and $r^H_{(j_b - i_b)}$ are learned offset embeddings for relative width $j_a - i_a$ and relative height $j_b - i_b$, respectively. The relative positional embeddings r^W and r^H are learned and shared across heads but not layers.

3.2. Shuffle attention module

Recently, the network architecture of neural pose regressors mainly contains two stages, i.e., a feature extractor and a global average pooling followed by a fully connected regressor. These works take advantage of transfer learning from image classification tasks, but in addition to benefit, pose regressor should be more robust to noise and perceive spatial information, which is confused by global pooling. Considering these two factors, we introduce a SA module [30] that can adjust the importance of each sub-feature by generating attention factors for each spatial location in each group, so that each individual group can autonomously enhance its learning expression and suppress possible noise.

As shown in Fig. 3, for a given $X \in R^{C\times H\times W}$, we divide X into G groups along the channel dimension, denoted by $X = [X_1, \ldots, X_G], X_k \in R^{C/G\times H\times W}$. Each feature map will be segmented along the channel dimension into two branches, denoted by $X_{k1}, X_{k2} \in R^{C/2G\times H\times W}$. One branch is used to generate channel attention feature maps by acquiring the inter-relationship of channels, while the other branch generates spatial attention maps via the analysis of space location relationships.

The output of the channel attention can be obtained by

$$X'_{k1} = \sigma\left[w_1 * \left(\frac{1}{H\times W}\sum_{i=1}^{H}\sum_{j=1}^{W} X_{k1(i,j)}\right) + b_1\right] * X_{k1} \tag{4}$$

Where $\frac{1}{H\times W}\sum_{i=1}^{H}\sum_{j=1}^{W} X_{k1(i,j)}$ represents the GAP operation. $w_1 \in R^{C/2G\times 1\times 1}$ and $b_1 \in R^{C/2G\times 1\times 1}$ are the two dynamic parameters to scale the feature map, (i, j) refers to the specific spatial position in input features, and σ is the sigmoid activation. As a result, the final output of spatial attention branch is followed by

$$X'_{k2} = \sigma\left(w_2 * GN\left(X_{k2}\right) + b_2\right) \cdot X_{k2} \tag{5}$$

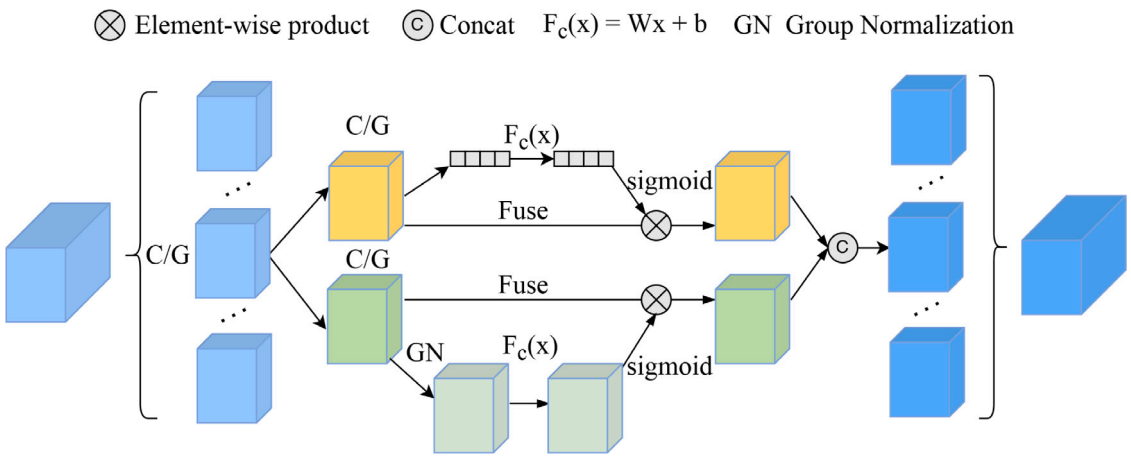


Fig. 3. An overview of the SA Module.

To obtain the final weighted feature map, the representative features obtained by the spatial and channel attention branches require to be aggregated with concatenative links, denoted by $X_k = concat\left[X'_{k1}, X'_{k2}\right]$. Then, we use the channel shuffle operation to allow cross-group information to flow along the channel dimension, which results in a more discriminative feature.

The attention block is flexible and can be inserted anywhere in network. To optimize the localization performance, we place the attention block after the Feature Encoder. This allows the network to focus on and highlight the most critical components.

3.3. Camera pose loss

Camera pose regression was first suggested by [8]. The proposed network architecture, named PoseNet, was based on a GoogLeNet backbone where the classifier was replaced with fully-connected layers to regress the absolute pose of the input image. Therefore, we add two fully connected layers at the end which are the final pose regressors for position and orientation. The pose regressors map the attention guided features $SA(x)$ to location $t \in R^3$ and orientation $r \in R^4$ respectively through Multilayer Perceptrons (MLPs) [38]:

$$[t, r] = MLPs\left(SA\left(f_{encoder}(I)\right)\right) \tag{6}$$

Camera pose regressors are trained to minimize the deviation between the ground truth pose $p^* =< t^*, r^* >$ and the predicted camera pose $p = \langle t, r \rangle$. The parameters inside the neural networks are optimized with L1 Loss to minimize a Pose Loss function [10] for an image I:

$$Loss(I) = \left\|t - t^*\right\|_1 e^{-\beta} + \beta + \left\|\log r - \log r^*\right\|_1 e^{-\gamma} + \gamma \tag{7}$$

Where β and γ are learned parameters controlling the balance between the two loss. The logarithm of a unit quaternion $\log r$ is the logarithm of a unit quaternion r. During training, β and γ are not fixed, we optimize them as parameters with an initialization β_0 and γ_0 for all scenes. Experimental details of β and γ are given in Section 4.4.

4. Experiments

Our proposed scheme focuses on improving the accuracy of absolute pose regression. Therefore, we evaluate it on multiple contemporary datasets used for benchmarking camera pose regressors, and compare the results with the recent state-of-the-art regression based absolute localization methods. We evaluated our method on 7Scenes [39] and Oxford RobotCar [40] datasets, which are commonly used to evaluate pose regression tasks.

4.1. Datasets

4.1.1. 7Scenes

The dataset is consisted of seven different indoor scenes captured by a handheld Kinect RGB-D camera, with a spatial extent of a few squared meters. All pictures were taken in a small-scale indoor office environment with a resolution of 640 × 480 pixels. Each scene contains two to seven sequences and each sequence has 500 or 1000 frames. The sequences contained in this dataset are recorded in different camera motion status and different conditions, such as the frames exhibit ambiguities (e.g. repeated steps in 'Stairs'), reflections (e.g. reflections in 'Kitchen'), motion blur, lighting conditions, texture-less surfaces, and variations in view point and trajectory.

4.1.2. Oxford RobotCar

The dataset contains over 100 continuous routes (about 10 kilometers) through central Oxford captured twice a week over a period of over a year. Therefore, the video sequences are full of complex traffic conditions and abundant with movable objects such as parked/moving vehicles, cyclists and pedestrians. It makes the dataset particularly challenging for vision-based localization tasks. In order to make a fair comparison, we extracted a subset from the whole data set of different scenes for experiments: FULL, with a total length of 9562 m. More details about these two sequences are provided in Table 3.

4.2. Implementation details

We implemented our algorithms with PyTorch, using the Adam optimizer [45] and initial learning rate of $\lambda = 5 \times 10^{-5}$. 256 × 256 pixels images are cropped for our network during both the training and testing phase with random and central cropping strategy respectively. The network is trained on a NVIDIA GeForce RTX 2070 GPU with the following hyperparameters: mini-batch size of 24, dropout rate probability of 0.5 and initializations of $\beta_0 = 0.0$ and $\gamma_0 = -3.0$. For the training on Oxford RobotCar dataset, random Color Jitter is additionally applied when data augmentation is performed, with brightness, contrast, and saturation set to a value of 0.7, and hue set to a value of 0.5 [38]. We note that this augmentation step is critical to improve the generalization ability of model over time conditions and various weather.

4.3. Comparative analysis of camera pose regressors

4.3.1. Experiments on 7Scenes dataset

Table 2 shows the quantitative results obtained when evaluating our model on 7Scenes dataset. We compared its performance with all pose regressors reported on this benchmark with a single image input. Our

Table 2
Localization results for the 7Scenes dataset (indoor localization). We report the median position/orientation error in meters/degrees for each method. The best results are highlighted in bold.

Method	Chess	Fire	Heads	Office	Pumpkin	Kitchen	Stairs	Avg
PoseNet [8]	0.32/8.12	0.47/14.4	0.29/12.0	0.48/7.68	0.47/8.42	0.59/8.64	0.47/13.8	0.45/9.94
PN-Learnable [10]	0.14/4.50	0.27/11.8	0.18/12.1	0.20/5.77	0.25/4.82	0.24/5.52	0.37/10.6	0.24/**7.87**
LSTM-PN [13]	0.24/5.77	0.34/11.9	0.21/13.7	0.30/8.08	0.33/7.0	0.37/8.83	0.40/13.7	0.31/9.85
BayesianPN [20]	0.37/7.24	0.43/13.7	0.31/**12.0**	0.48/8.04	0.61/7.08	0.58/7.54	0.48/13.1	0.47/9.81
GPoseNet [21]	0.20/7.11	0.38/12.3	0.21/13.8	0.28/8.83	0.37/6.94	0.35/8.15	0.37/12.5	0.31/9.95
BranchNet [34]	0.18/5.17	0.34/**8.99**	0.20/14.15	0.30/7.05	0.27/5.10	0.33/7.40	0.38/**10.26**	0.29/8.30
MLFBPPose [41]	0.12/5.82	0.26/11.99	**0.14**/13.54	**0.18**/8.24	0.21/7.05	0.22/8.14	0.38/10.26	0.22/9.29
Bui et al. [42]	0.12/4.8	0.29/10.2	0.15/12.0	0.21/6.6	0.22/6.5	0.26/5.8	0.30/12.2	0.22/8.3
ViPR [43]	0.22/7.89	0.38/12.74	0.21/16.41	0.35/9.59	0.37/8.45	0.40/9.32	0.31/12.65	0.32/11.01
IRPNet [44]	0.13/5.64	0.25/9.67	0.15/13.1	0.24/6.33	0.22/5.78	0.30/7.29	0.34/11.6	0.23/8.49
TransBoNet	**0.11**/**4.48**	**0.25**/12.46	0.18/14.00	0.20/**5.08**	**0.19**/**4.77**	**0.17**/**5.35**	**0.30**/13.04	**0.20**/8.45

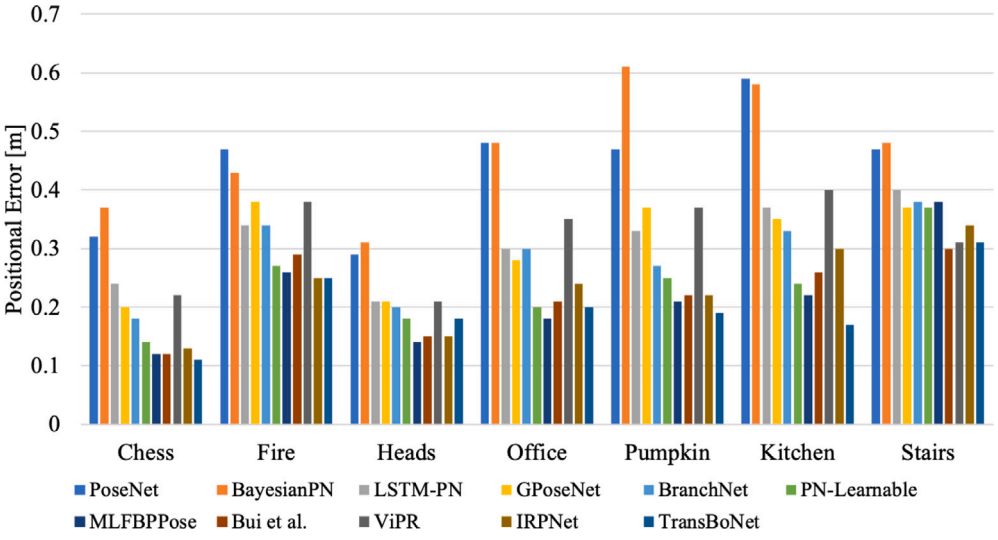


Fig. 4. Visual comparison of position media error on 7Scenes dataset.

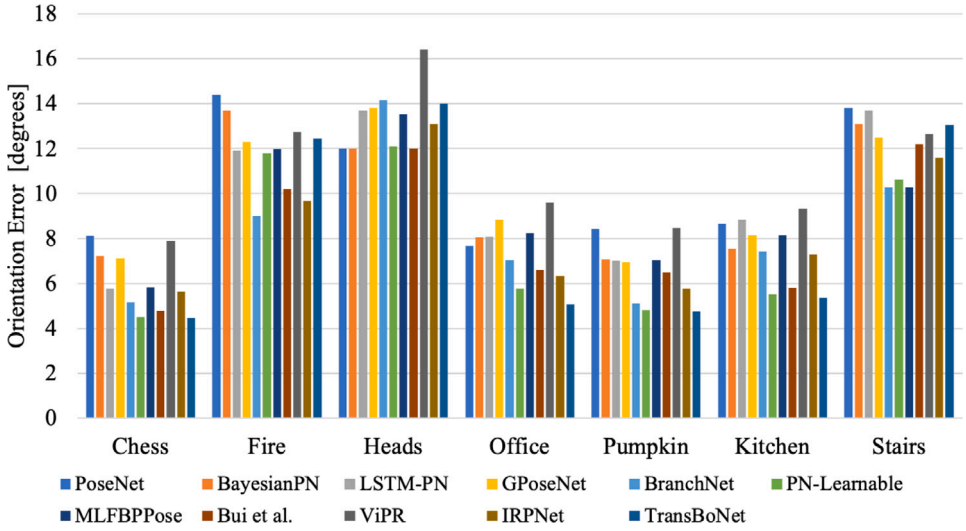
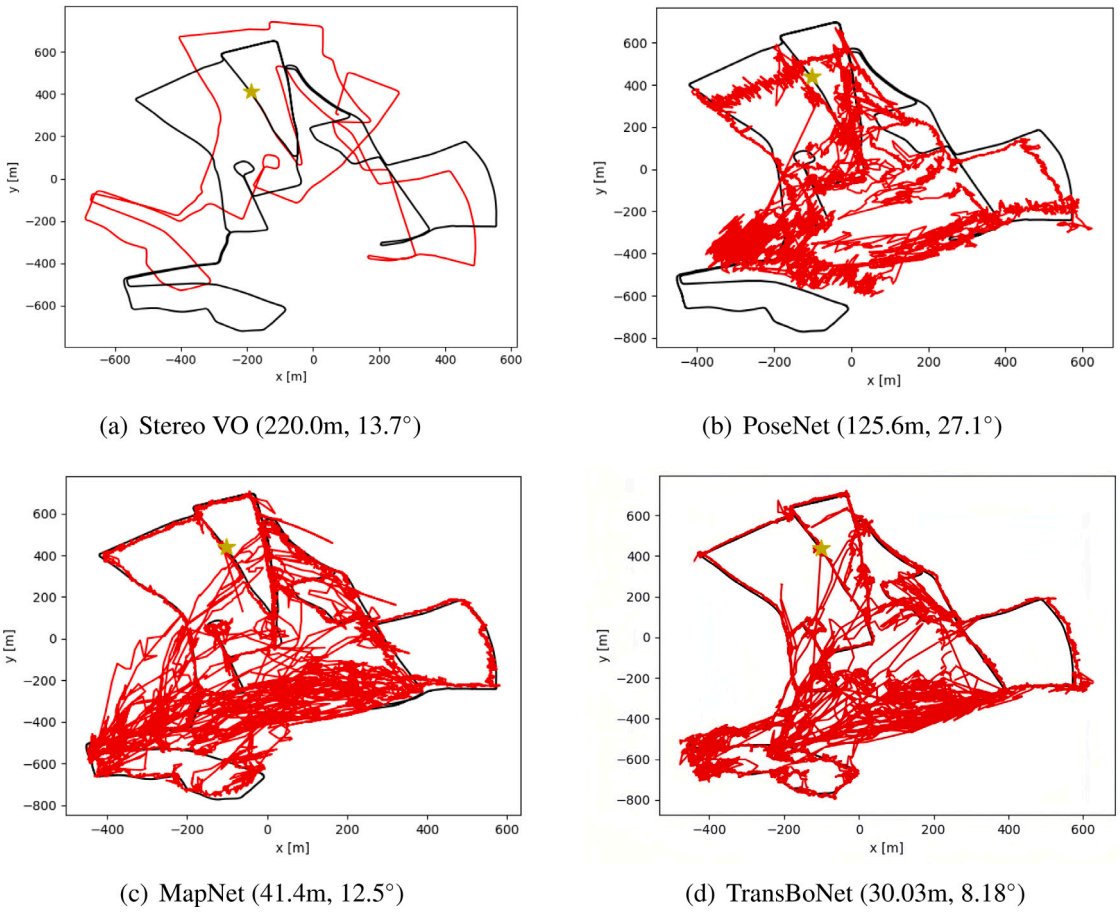


Fig. 5. Visual comparison of orientation media error on 7Scenes dataset.

model achieves the best positional error in five of the seven scenes, and achieves the highest number of state-of-the-art results overall. Especially in scenes with texture-less regions (such as fire and pumpkin) and highly repetitive textures (such as hess). It reduces the position error from 0.47 m to 0.19 m and the rotation error from 8.42° to 4.77° in the Pumpkin scene, the position error from 0.32 m to 0.11 m and the rotation error from 8.12° to 4.48° in the chess scene, which is a significant improvement over prior art. The media localization errors of the 7Scenes dataset are shown in Figs. 4 and 5.

4.3.2. *Experiments on RobotCar dataset*

Table 4 shows the quantitative comparison of PoseNet [8,10,20], MapNet [12], LsG [46] and our method. Due to the complex environment of the RobotCar dataset and the long sequence length, using



(a) Stereo VO (220.0m, 13.7°) (b) PoseNet (125.6m, 27.1°)

(c) MapNet (41.4m, 12.5°) (d) TransBoNet (30.03m, 8.18°)

Fig. 6. Results of FULL1 scenes of the Oxford RobotCar dataset. The black and red lines indicate ground truth and predicted trajectories respectively. The stars in the trajectory represent the starting point. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

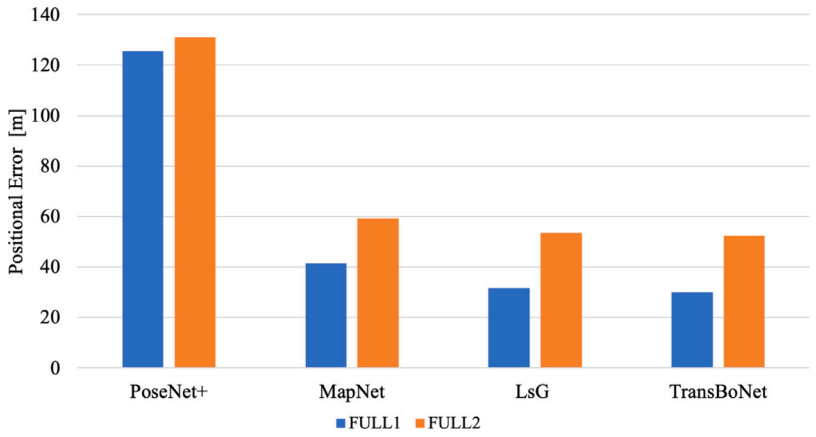


Fig. 7. Visual comparison of position mean error on RobotCar dataset.

Table 3
Training and testing Sequences of Oxford RobotCar.

Sequence	Time	Mode
–	2014-11-28-12-07-13	Training
–	2014-12-02-15-30-08	Training
FULL1	2014-12-09-13-21-02	Testing
FULL2	2014-12-12-10-45-15	Testing

the mean error can more fully reflect the stability of the model. Since the training and testing sequences are captured under different conditions at different times, PoseNet can be difficult to deal with these changes and outputs inaccurate estimations of a large number of outliers. MapNet generates more accurate results and reduces many outliers by introducing relative poses between consecutive frames as additional constraints. However, the larger areas may contain more locally similar appearances, reducing the ability of the localization systems. By employing content enhancement, LsG improves this problem to a certain extent, but it reduces the accuracy. In contrast, considering content and movement, our model addresses these challenges more effectively. The mean localization errors of the Oxford RobotCar dataset are shown in Figs. 7 and 8.

Table 4
Camera localization results on the FULL of the Oxford RobotCar. For each sequence, we calculate the median and mean position/orientation error in meters/degrees for various methods. The best results are highlighted.

Sequence	PoseNet+ [8,10,20]		MapNet [12]		LsG [46]		TransBoNet	
–	Mean	Median	Mean	Median	Mean	Median	Mean	Median
FULL1	125.6/27.1	107.6/22.5	41.4/12.5	17.94/6.68	31.65/**4.51**	–	**30.03**/8.18	**6.46/1.25**
FULL2	131.1/26.05	101.8/20.1	59.3/14.8	20.04/6.39	53.45/**8.60**	–	**52.42**/10.98	**8.77/1.69**
Average	128.35/26.58	104.7/21.3	50.35/13.6	18.99/6.54	42.55/**6.56**	–	**41.23**/9.58	**7.62/1.47**

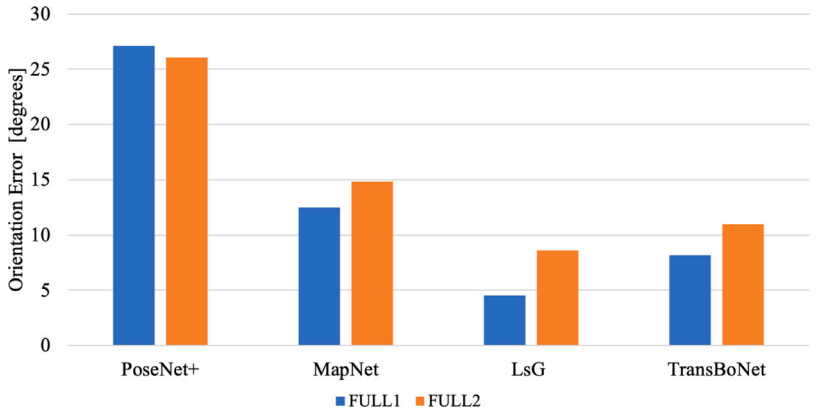


Fig. 8. Visual comparison of orientation mean error on RobotCar dataset.

Fig. 9. Heatmap visualization of attention weights from Oxford RobotCar generated from the original image (top) and with attention model (bottom). TransBoNet learns to ignore visually uninformative features, such as roads in the above figure, and instead focus on more unique objects, such as distant skylines. It also learns to reject some dynamic objects, such as vehicles in the middle figure and pedestrians in the bottom figure, so as to achieve more robust camera localization.

We studied the reasons why our model was significantly better than the baseline on Oxford RobotCar dataset. Fig. 6 shows the trajectories predicted by Stereo VO, PoseNet, MapNet and our model in FULL1 scene. Stereo VO is the official baseline of Oxford RobotCar. As shown in Fig. 6, the Stereo VO generates very smooth trajectories because the relative poses between consecutive frames can be accurately calculated. However, with the increase of route length, it will drift significantly. PoseNet predicts many outliers far from the ground truth locations. Using the motion consistency, MapNet improves the accuracy of predicted poses and greatly reduces the number of outliers. Yet, due to the local similarities of observed values in large scenes, a large number of outliers still exist throughout the region. As shown in Fig. 9, we adopt attention to guide our model to automatically focus on some static and stable objects/areas. These areas are closely related to the potential geometric features of the environment, which enables robust pose estimation in the wild. Compared to PoseNet and MapNet, our model eliminates outliers more effectively and has more stable performance in noisy regions generated by the previous two methods.

Table 5
We calculate the median and mean position/orientation errors (in meters/degrees) on 7Scenes and Oxford RobotCar while β and γ are set as different values, respectively. The best results are highlighted.

Scene	$\beta = 0.0,\ \gamma = -2.0$	$\beta = 0.0,\ \gamma = -3.0$	$\beta = 0.0,\ \gamma = -4.0$	$\beta = 1.0,\ \gamma = -3.0$
		7Scenes		
Chess	0.11/4.66	**0.11/4.48**	0.12/4.86	0.11/4.54
Fire	0.28/13.38	**0.25/12.46**	0.26/12.96	0.25/13.14
Heads	0.18/14.66	0.18/**14.00**	0.18/14.93	**0.17**/14.43
Office	0.21/6.67	**0.20/5.08**	0.20/7.04	0.21/6.73
Pumpkin	0.19/5.97	**0.19/4.77**	0.20/5.72	0.19/5.36
Kitchen	0.18/5.83	**0.17/5.35**	0.19/6.22	0.19/5.65
Stairs	**0.28**/12.75	**0.30**/12.04	0.29/14.02	0.30/13.02
		Oxford RobotCar		
FULL1	39.61/**7.75**	**30.03**/8.18	35.42/8.22	38.64/8.71
FULL2	59.71/12.22	**52.42/10.98**	58.62/11.35	57.35/12.74

Table 6
Ablation study of our model on 7Scenes and Oxford RobotCar. The model (Basic) denotes the model without using any attention mechanism. We calculate the median and mean position/orientation errors (in meters/degrees) on 7Scenes and Oxford RobotCar, respectively. The best results are highlighted.

Scene	Basic	Basic+TBo	Basic+TBo+SA
		7Scenes	
Chess	0.14/7.15	0.14/6.21	**0.11/4.48**
Fire	0.34/13.31	0.30/**12.88**	**0.25**/13.33
Heads	0.21/15.07	0.18/**13.45**	**0.18**/14.00
Office	0.25/8.68	0.21/8.67	**0.20/7.08**
Pumpkin	0.26/7.20	0.25/7.89	**0.19/5.77**
Kitchen	0.19/8.56	0.18/7.73	**0.17/5.35**
Stairs	0.42/13.63	0.38/12.03	**0.31/13.52**
Average	0.26/10.51	0.23/9.84	**0.20/9.04**
		Oxford RobotCar	
FULL1	50.93/14.52	38.27/10.16	**31.03/8.18**
FULL2	85.63/20.21	61.35/13.58	**53.42/10.98**
Average	68.28/17.37	49.81/11.87	**42.43/9.58**

4.4. Sensitivity study of β and γ

The hyperparameters β and γ are learned parameters controlling the balance between the two loss. In the camera pose regression task, quaternion is widely used to denote the orientation due to it is easy to be continuously differentiable. As we expect quaternion values to have much smaller values, their noise, γ should be much smaller than the position noise, β, which can be many meters in magnitude. Since γ should be much smaller than β, orientation should be weighted much higher than position [10].

We perform the sensitivity study to see whether the performance varies much with the changes of the hyperparameter β and γ. Specifically, we set β and γ to values chosen from the predefined ranges, train model with these hyperparameter values on the training set, and then report pose prediction error on the test set.

We find that this loss (Eq. (7)) is very robust to initial choice for the homoscedastic task uncertainty values [10]. Only an approximate initial guess is required, GeoPoseNet [10], MapNet [12] and AttLoc [38], the initial values of beta and gamma are both set to 0.0 and −3.0, respectively, for all scenes. As a baseline, we validate this initial set of values on a representative 7Scenes and Oxford RobotCar dataset. From Table 5, we can see that the localization error does not change much, but is consistently higher than the baseline. Therefore, we choose $\beta = 0.0$ and $\gamma = -3.0$ as initial values.

4.5. Ablation study

We conduct an ablation study to validate the effectiveness on the introduced attention module above 7scenes and Oxford RobotCar datasets. As shown in Table 6, compared with a basic version without the attention module, and a Transformer Bottleneck (TBo) version with the hybrid attention. The comparison of the Basic model and our model shows that the model performance significantly increases on both datasets by introducing the attention into the whole pose regression model: it shows a 23% improvement in position accuracy and 14% in rotation accuracy on the 7Scenes dataset; On Oxford RobotCar dataset, the position accuracy is improved by 10% and the rotation accuracy is improved by 11%.

In order to evaluate the performance of our proposed TBo module, we analyze the localization accuracy of two models: Basic and Basic+TBo. It can be seen that the TBo model has obtained the best localization performance. Therefore, it can be well applied to the pose regression network. In addition, we also compare the SA module introduced in the pose regression process, and find that the SA module further improves the localization performance of the network.

5. Conclusion

Camera localization is a challenging task in computer vision due to scene dynamics and the highly variable appearance of the environment. In this paper, we present a hybrid attention-based method for fast and accurate camera localization from only RGB images. We propose a novel visual localization framework, which significantly improves the robustness and accuracy of pose regressors in dynamic environments, especially in complex urban traffic. The TBo block can be easily incorporated into modern neural networks to improve feature extraction in dynamic scenes. The introduced shuffle attention can encourage the network to learn geometrically robust features, reducing the impacts from dynamic objects and changing illumination. Experiments show that the proposed localization method can significantly improve the robustness and accuracy on both outdoor and indoor datasets, especially on the challenging RobotCar dataset. However, considering that the method of directly regressing the camera pose has low accuracy, in future work, we will try to refine the attention module and determine whether the localization performance of the camera localization method based on scene coordinate regression can be improved in challenging large-scale outdoor scenes. 6DoF pose of a given image is calculated by a PnP-RANSAC scheme instead of the pose regressor.

Declaration of competing interest

We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work, there is no professional or other personal interest of any nature or kind in any product, service and/or company that could be construed as influencing the position presented in, or the review of, the manuscript entitled.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by National Defense Advanced Research Program of China under Grant No. 61405170206 and Key Research and Development Program of Shaanxi Province of China under Grant No. 2022GY-245.

References

[1] R. Munoz-Salinas, R. Medina-Carnicer, UcoSLAM: Simultaneous localization and mapping by fusion of KeyPoints and squared planar markers, Pattern Recognit. 101 (2020) 107193.
[2] Y. Fan, Q. Zhang, Y. Tang, S. Liu, H. Han, Blitz-SLAM: A semantic SLAM in dynamic environments, Pattern Recognit. 121 (2022) 108225.
[3] R. Muñoz-Salinas, M.J. Marín-Jimenez, E. Yeguas-Bolivar, R. Medina-Carnicer, Mapping and localization from planar markers, Pattern Recognit. 73 (2018) 158–171.

[4] N. Piasco, D. Sidibé, C. Demonceaux, V. Gouet-Brunet, A survey on visual-based localization: On the benefit of heterogeneous data, Pattern Recognit. 74 (2018) 90–109.
[5] T. Sattler, Q. Zhou, M. Pollefeys, L. Leal-Taixé, Understanding the limitations of CNN-based absolute camera pose regression, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 3297–3307.
[6] J. Wang, Y. Qi, Visual camera relocalization using both hand-crafted and learned features, Pattern Recognit. (2023) 109914.
[7] C. Celik, H.S. Bilge, Content based image retrieval with sparse representations and local feature descriptors : A comparative study, Pattern Recognit. 68 (2017) 1–13.
[8] A. Kendall, M. Grimes, R. Cipolla, PoseNet: A convolutional network for real-time 6-DOF camera relocalization, in: 2015 IEEE International Conference on Computer Vision, 2015, pp. 2938–2946.
[9] I. Melekhov, J. Ylioinas, J. Kannala, E. Rahtu, Image-based localization using hourglass networks, in: 2017 IEEE International Conference on Computer Vision Workshops, 2017, pp. 870–877.
[10] A. Kendall, R. Cipolla, Geometric loss functions for camera pose regression with deep learning, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6555–6564.
[11] R. Clark, S. Wang, A. Markham, N. Trigoni, H. Wen, VidLoc: A deep spatio-temporal model for 6-DoF video-clip relocalization, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2652–2660.
[12] S. Brahmbhatt, J. Gu, K. Kim, J. Hays, J. Kautz, Geometry-aware learning of maps for camera localization, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 2616–2625.
[13] F. Walch, C. Hazirbas, L. Leal-Taixé, T. Sattler, S. Hilsenbeck, D. Cremers, Image-based localization using LSTMs for structured feature correlation, in: 2017 IEEE International Conference on Computer Vision, 2017, pp. 627–637.
[14] L. Liu, H. Li, Y. Dai, Efficient global 2D-3D matching for camera localization in a large-scale 3D map, in: 2017 IEEE International Conference on Computer Vision, 2017, pp. 2391–2400.
[15] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, J. Sivic, NetVLAD: CNN architecture for weakly supervised place recognition, IEEE Trans. Pattern Anal. Mach. Intell. 40 (6) (2018) 1437–1451.
[16] U. Nadeem, M. Bennamoun, R. Togneri, F. Sohel, A. Miri Rekavandi, F. Boussaid, Cross domain 2D-3D descriptor matching for unconstrained 6-DOF pose estimation, Pattern Recognit. 142 (2023) 109655.
[17] P. Zhang, C. Zhang, B. Liu, Y. Wu, Leveraging local and global descriptors in parallel to search correspondences for visual localization, Pattern Recognit. 122 (2022) 108344.
[18] Q. Li, R. Cao, K. Liu, Z. Li, J. Zhu, Z. Bao, X. Fang, Q. Li, X. Huang, G. Qiu, Structure-guided camera localization for indoor environments, ISPRS J. Photogramm. Remote Sens. 202 (2023) 219–229.
[19] S. Wang, R. Clark, H. Wen, N. Trigoni, End-to-end, sequence-to-sequence probabilistic visual odometry through deep neural networks, Int. J. Robot. Res. 37 (2018) 513–542.
[20] A. Kendall, R. Cipolla, Modelling uncertainty in deep learning for camera relocalization, in: 2016 IEEE International Conference on Robotics and Automation, 2016, pp. 4762–4769.
[21] M. Cai, C. Shen, I.D. Reid, A hybrid probabilistic model for camera relocalization, in: Proceedings of the 29th British Machine Vision Conference, 2019, pp. 1–12.
[22] Z. Huang, Y. Xu, J. Shi, X. Zhou, H. Bao, G. Zhang, Prior guided dropout for robust visual localization in dynamic environments, in: 2019 IEEE/CVF International Conference on Computer Vision, 2019, pp. 2791–2800.
[23] M. Sarıgül, L. Karacan, Region contrastive camera localization, Pattern Recognit. Lett. 169 (2023) 110–117.
[24] E. Parisotto, D.S. Chaplot, J. Zhang, R. Salakhutdinov, Global pose estimation with an attention-based recurrent network, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2018, pp. 350–359.
[25] J. Hu, L. Shen, S. Albanie, G. Sun, E. Wu, Squeeze-and-excitation networks, IEEE Trans. Pattern Anal. Mach. Intell. 42 (8) (2020) 2011–2023.
[26] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, Q. Hu, ECA-net: Efficient channel attention for deep convolutional neural networks, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 11531–11539.
[27] X. Wang, R. Girshick, A. Gupta, K. He, Non-local neural networks, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 7794–7803.
[28] Y. Cao, J. Xu, S. Lin, F. Wei, H. Hu, GCNet: Non-local networks meet squeeze-excitation networks and beyond, in: 2019 IEEE/CVF International Conference on Computer Vision Workshop, 2019, pp. 1971–1980.
[29] S. Woo, J. Park, J.-Y. Lee, I.S. Kweon, CBAM: Convolutional block attention module, in: European Conference on Computer Vision, 2018, pp. 3–19.
[30] Q.-L. Zhang, Y.-B. Yang, SA-Net: Shuffle attention for deep convolutional neural networks, in: 2021 IEEE International Conference on Acoustics, Speech and Signal Processing, 2021, pp. 2235–2239.
[31] G. Chen, J. Lu, M. Yang, J. Zhou, Spatial-temporal attention-aware learning for video-based person re-identification, IEEE Trans. Image Process. 28 (9) (2019) 4192–4205.
[32] X. Li, X. Hu, J. Yang, Spatial group-wise enhance: Improving semantic feature learning in convolutional networks, 2019, arXiv e-prints arXiv:1905.09646.
[33] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, H. Lu, Dual attention network for scene segmentation, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 3141–3149.
[34] J. Wu, L. Ma, X. Hu, Delving deeper into convolutional neural networks for camera relocalization, in: 2017 IEEE International Conference on Robotics and Automation, 2017, pp. 5644–5651.
[35] P. Shaw, J. Uszkoreit, A. Vaswani, Self-attention with relative position representations, in: 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2018, pp. 464–468.
[36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017, pp. 6000–6010.
[37] A. Srinivas, T.-Y. Lin, N. Parmar, J. Shlens, P. Abbeel, A. Vaswani, Bottleneck transformers for visual recognition, in: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 16514–16524.
[38] B. Wang, C. Chen, C. Lu, P. Zhao, N. Trigoni, A. Markham, Atloc: Attention guided camera localization, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2020, pp. 10393–10401.
[39] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, A. Fitzgibbon, Scene coordinate regression forests for camera relocalization in RGB-D images, in: 2013 IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 2930–2937.
[40] W. Maddern, G. Pascoe, C. Linegar, P. Newman, 1 year, 1000 km: The oxford RobotCar dataset, Int. J. Robot. Res. 36 (2017) 3–15.
[41] X. Wang, X. Wang, C. Wang, X. Bai, E.R. Hancock, Discriminative features matter: Multi-layer bilinear pooling for camera localization, in: 30th British Machine Vision Conference, 2020, pp. 1–12.
[42] M. Bui, C. Baur, N. Navab, S. Ilic, S. Albarqouni, Adversarial networks for camera pose regression and refinement, in: 2019 IEEE/CVF International Conference on Computer Vision Workshop, 2019, pp. 3778–3787.
[43] F. Ott, T. Feigl, C. Löffler, C. Mutschler, Vipr: Visual-odometry-aided pose regression for 6dof camera localization, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2020, pp. 187–198.
[44] Y. Shavit, R. Ferens, Do we really need scene-specific pose encoders? in: 25th International Conference on Pattern Recognition, 2021, pp. 3186–3192.
[45] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: 3rd International Conference on Learning Representations, 2015, pp. 1–15.
[46] F. Xue, X. Wang, Z. Yan, Q. Wang, J. Wang, H. Zha, Local supports global: Deep camera relocalization with sequence enhancement, in: 2019 IEEE/CVF International Conference on Computer Vision, 2019, pp. 2841–2850.

Xiaogang Song received the B.S. degree in computer science from Xi'an University of Technology, China, in 2008, M.S. and Ph.D. degrees in computer science from Northwestern Polytechnical University, China, in 2011 and 2018 respectively. He is currently an Associate Professor and Master Supervisor at the Xi'an University of Technology. He is a member of China Computer Federation, Intelligent Robot Committee of the Computer Federation, IEEE, Shaanxi Embedded Committee, ACM Editorial Committee. His research interests include computer vision and autonomous navigation.

Hongjuan Li received the B.S. degree from Tianjin University of Commerce, Tianjin, China, in 2020. She is currently a postgraduate student in the University of Xi'an University of Technology. Her research interests include indoor localization, image localization and vison-based SLAM.

Li Liang received the M.S. degree from Xi'an University of Technology, Xi'an, China, in 1991. She is currently an Associate Professor and Master Supervisor at the Xi'an University of Technology. Supports or participates in more than 20 vertical and horizontal scientific research projects. Her research interests include computer vision, deep learning and intelligent control.

Weiwei Shi received the M.S. degree in applied mathematics and the Ph.D. degree in pattern recognition and intelligent system from Xi'an Jiaotong University, Xi'an, China, in 2012 and 2019, respectively. In 2021, he won one second prize of Shaanxi University Science and Technology Award (the second person to complete). Served as the review expert of the National Natural Science Foundation of China. His current research interests include multimedia analysis, machine learning, and pattern recognition.

Guo Xie received the B.S. and M.S. degrees from the Xian University of Technology, Xian, China, in 2005 and 2008, respectively, and the D.E. degree from Nihon University, Tokyo, Japan, in 2013. He is currently a Professor with the Xian University of Technology. He was awarded the title of the Hundred-Talent Program of Shaanxi Province. His research interests include intelligent information processing and intelligent transportation.

Xiaofeng Lu received the B.S and M.S. degrees in computer science and technology from the Xi'an University of Technology, Xi'an, China, in 2001 and 2006, respectively, and the Ph.D. degree from Nihon University, Tokyo, Japan, in 2014. Since 2014, he has been with the School of Computer Science and Engineering, Xi'an University of Technology, Xi'an, where he became an Associate Professor. His current research interests include intelligent transport systems, pattern recognition, and image processing.

Xinhong Hei received the Ph.D. degree from Nihon University, Japan, in 2008. He is currently a Professor and Doctoral Supervisor at the Xi'an University of Technology. He is a member of ACM, IEEE, China Computer Federation, Network Information Services Committee of China Automation Federation. He was awarded the title of Shaanxi Province Youth Science and technology innovation leader. His research interests include artificial intelligence, computer vision, large data and their applications in rail transit systems.