EpiLoc: Deep Camera Localization Under Epipolar Constraint

Luoyuan Xu[1], Tao Guan[1*], Yawei Luo[2], Yuesong Wang[1], Zhuo Chen[1], WenKai Liu[1]
[1] School of Computer Science & Technology, Huazhong University of Science & Technology
[e-mail: xu_luoyuan@hust.edu.cn,qd_gt@hust.edu.cn,yuesongwang@hust.edu.cn,
cz_007@hust.edu.cn, wenkai_liu@hust.edu.cn]
[2] School of Computer Science & Technology, Zhejiang University
[e-mail: yaweiluo@zju.edu.cn]
*Corresponding author: Tao Guan

*Received November 30, 2021; revised April 16, 2022; accepted May 9, 2022;
published June 30, 2022*

Abstract

Recent works have shown that the geometric constraint can be harnessed to boost the performance of CNN-based camera localization. However, the existing strategies are limited to imposing image-level constraint between pose pairs, which is weak and coarse-gained. In this paper, we introduce a pixel-level epipolar geometry constraint to vanilla localization framework without the ground-truth 3D information. Dubbed EpiLoc, our method establishes the geometric relationship between pixels in different images by utilizing the epipolar geometry thus forcing the network to regress more accurate poses. We also propose a variant called EpiSingle to cope with non-sequential training images, which can construct the epipolar geometry constraint based on a single image in a self-supervised manner. Extensive experiments on the public indoor 7Scenes and outdoor RobotCar datasets show that the proposed pixel-level constraint is valuable, and helps our EpiLoc achieve state-of-the-art results in the end-to-end camera localization task.

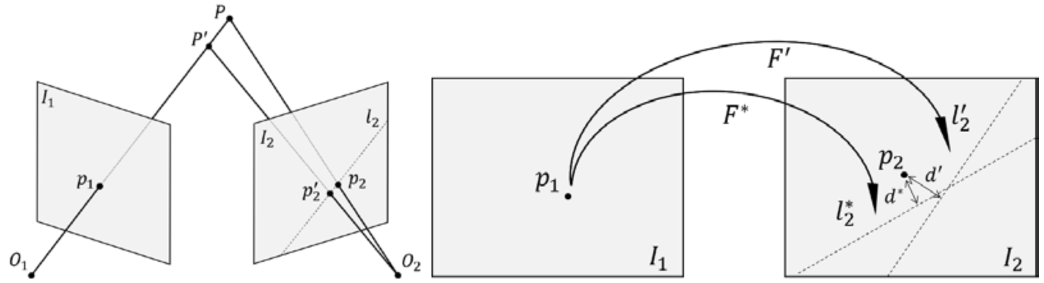
Keywords: Camera localization, End-to-end, Epipolar geometry, Pixel-level constraint.

http://doi.org/10.3837/tiis.2022.06.014 **ISSN : 1976-7277**

1. Introduction

Camera localization that recovers the 3D translation and rotation from a single image is one of the fundamental tasks in a wide variety of applications, *e.g.*, autonomous driving, robotics, and augmented reality. Camera localization has been studied for decades and a number of structure-based approaches have been proposed [19],[24],[26],[23]. Generally, these structure-based methods rely on a sparse map consisting of the 2D keypoints of query images, the 3D points, as well as the visible relationship between them. In the general pipeline, the matches between 2D and 3D points are established by comparing their descriptors, then the set of 2D-3D correspondences will be used to recover the poses of the query image by a Perspective-n-Point solver in a RANSAC loop [8]. By leveraging 3D information or geometry of the scene, the structure-based methods can yield precise poses, but at a price of large memory footprint and computational cost.

Motivated by the success of deep learning in a variety of computer vision tasks, such as image classification [21],[28], object detection [27],[32], and semantic segmentation [39],[2], researchers are taking step towards exploiting Deep Neural Networks (DNNs) or Recurrent Neural Networks (RNNs) to regress global poses in an end-to-end supervised manner. The seminal PoseNet [16] enables us to directly regress the absolute pose from one image with a simpler pipeline, shorter inference times, and lower memory footprint than structure-based methods but resulting in larger localization error in both indoor and outdoor scenarios. One possible cause of performance degradation is the lack of geometric information during the network training. To overcome this drawback, MapNet [3] proposes a geometry-aware learning paradigm, additionally enforcing geometric constraint between pose predictions for image pairs. Although achieving better results, the image-level geometric constraint between poses can be only regarded as coarse-grained supervision, while other stronger geometric constraint is not guaranteed to be considered in MapNet [3].



(a) The epipolar geometry **(b) Minimizing epipolar distance**

Fig. 1. The epipolar geometry constraint between input images I_1 and I_2 restricts the corresponding pixel p_2 of a query pixel p_1 to fall on its epipolar line I_2. Such constraint is often unsatisfied due to the inaccurately estimated pose and its derived fundamental matrix F'. Our proposed EpiLoc reinforces such pixel-level constraint by minimizing the point-to-line distance d' between p_2 and p'_2. The optimized fundamental matrix F^* can yield a smaller point-to-line distance d^*, indicating a more accurate camera pose that conforms with the epipolar geometry constraint.

In this paper, we propose to capitalize on a finer-grained geometric constraint as tailored supervision for the camera localization task. Specifically, we consider the epipolar geometric relationships between pixels as the constraint, where the corresponding pixel of a query pixel is supposed to fall on its epipolar line accordingly. However, such relationship can only be satisfied if the relative pose between images is accurately estimated. It motivates our proposed

EpiLoc to employ the epipolar geometric constraint as novel pixel-level supervision on the localization network. In a sequential data scenario, the pixel correspondences between images could be obtained by optical flow [13],[6], while the relative pose between images is derived from the estimated absolute pose of EpiLoc, as shown in **Fig. 1**. In order to apply our method to the non-sequential data scenario, we additionally propose EpiSingle which can construct the epipolar geometry constraint based on a single image in a self-supervised manner. EpiSingle computes the epipolar constraint between the estimated and ground truth pose of a single image without access to the corresponding relationship of pixels, which further boosts the applicability of EpiLoc. The main contributions of our paper are summarised as follows:

- Different from the coarse motion-geometry constraint used by previous methods, EpiLoc establishes the geometric relationship between pixels without the 3D model, which is a finer-grained geometric constraint and makes our model achieve accurate and robust pose estimation.
- As an extension of EpiLoc, we design EpiSingle for the non-sequential data scenario. EpiSingle computes the epipolar constraint between the estimated and ground truth pose of a single image without access to the corresponding relationship of pixels, which further boosts the applicability of EpiLoc.
- Extensive experiments on both indoor and outdoor datasets show the state-of-the-art results of our proposed method in the end-to-end camera localization task, which proves the effectiveness of epipolar geometric constraint.

The rest of the paper is structured as follows. Sec. 2 introduces the related work. Sec. 3 describes the epipolar geometry and the fundamental matrix. The details of our EpiLoc and EpiSingle are provided in Sec. 4. We show the experimental results in Sec. 5 and conclude in Sec. 6.

2. Related work

2.1 DNN-based camera localization in an end-to-end fashion

Recently, researchers leveraged DNN to recover camera pose directly from one image or multiple images. The pipeline of depth camera localization is simple and the inference times are short compared to the structure-based approach. The seminal DNN-based camera localization in an end-to-end fashion, PoseNet [16], is realized by a truncated GoogLeNet [27] followed by three FC layers. Some works improve localization accuracy by estimating the uncertainty of the predicted pose with Bayesian CNN [14] or replacing the encoder block with ResNet34 [22]. Some researchers proposed to utilize LSTM to spatially [30] and temporally [5] reduce localization errors. Kendall *et al.* [15] proposed to learn the weight between the translation and orientation loss and introduced a geometric reprojection error through 3D points. Moreover, the self-attention module is added to the pipeline and achieves good results [12],[21].

In addition to taking a single image as input as mentioned earlier, the researchers also proposed using multiple images [5],[3],[34]. MapNet [3] adds additional motion constraint *i.e.* relative pose between image pairs during training. LSG [34] adopts a convolutional LSTM [33] for an extra visual odometry estimation and employs a soft attention mechanism to remedy the finite capacity of recurrent units. Both of these works introduce coarse geometric information through motion constraints. In contrast, our work imposes the epipolar constraint on adjacent images to ensure that geometric constraint is satisfied between pixels of different images.

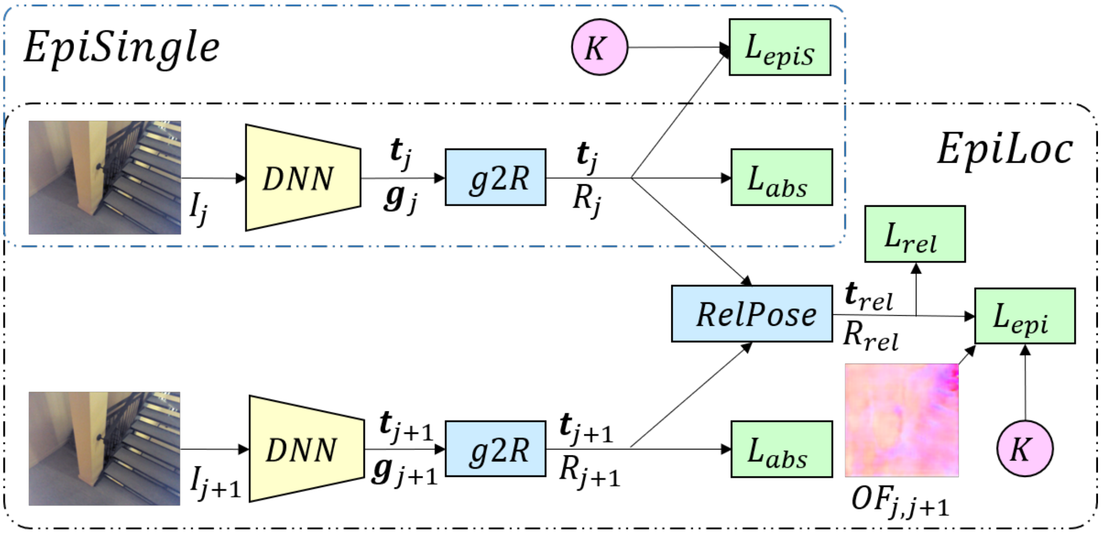


Fig. 2. Training pipeline of the Epi-architecture.
g2R and *RelPose* are the data processing functions, where *g2R* converts the logarithm of quaternion into a rotation matrix, and *RelPose* calculates the relative pose. *K* is the camera internal parameters. EpiLoc takes adjacent frames and the optical flow $\boldsymbol{OF_{j,j+1}}$ as input. EpiSingle takes one image as input.

2.2 Epipolar geometry as an additional constraint

The epipolar geometry constraint is the constraint between the pixels of two views, without the need for 3D structures. So many researchers leveraged the epipolar constraint on their tasks to achieve better results, such as 3D human pose estimation [18], depth estimation [38],[9],[36], neural rendering [29], keypoint detection [35], optical flow [37]. GLNet [4] is a system that combines multiple tasks, including depth estimation, optical flow, camera pose, and intrinsic parameters, which uses epipolar constraint to constrain the output of pose and optical flow modules and achieves a considerable performance gain. The pose modules of GLNet [4] output the relative pose of two images, but our work focuses on the end-to-end camera localization task, which regresses the absolute pose of a given image.

3. Preliminaries

3.1 Epipolar geometry

The epipolar geometry is the intrinsic projective geometry between two views, which is independent of the scene geometry structure but depends on the camera's internal parameters and relative pose [10]. As described in **Fig. 1**, the epipolar geometry constraint between two views $\boldsymbol{I_1}$ and $\boldsymbol{I_2}$ is expressed mathematically by the fundamental matrix $\boldsymbol{F}$, which is a 3x3 matrix with rank 2. To be specific, all corresponding pixels $\boldsymbol{p_1}, \boldsymbol{p_2}$ ($\boldsymbol{p_1}\in\boldsymbol{I_1}$, $\boldsymbol{p_2}\in\boldsymbol{I_2}$) satisfy the relation $\boldsymbol{p^T_2\ Fp_1 = 0}$. In our work, the pixel correspondences between adjacent frames are obtained by Flownet2 [13], which are accurate and dense.

3.2 The fundamental matrix

Given the rotation matrix $\boldsymbol{R_i}$, the camera center $\boldsymbol{O_i}$, the camera internal $\boldsymbol{K_i}$ for image $\boldsymbol{I_i}$, and the corresponding $\boldsymbol{R_j}$, $\boldsymbol{O_j}$ and $\boldsymbol{K_j}$ for image $\boldsymbol{I_j}$. Note that $\boldsymbol{R}$ is the rotation matrix from world to camera coordinates. The relative rotation and translation from image $\boldsymbol{I_i}$ to $\boldsymbol{I_j}$ are $\boldsymbol{R_{ji} = R_jR^{-1}_i}$, $\boldsymbol{t_{ji}}$

= R**$_j$**(O**$_i$**-O**$_j$**). The fundamental matrix ***F**$_{ji}$* from image ***I**$_i$* to ***I**$_j$* could be calculated by:

$$F_{ji} = K_j^{-T} \left[t_{ji} \right]^{\wedge} R_{ji} K_i^{-1} \tag{1}$$

where [·]^ is a skew-symmetric symbol, which can transform a 3d vector into a skew-symmetric matrix. Our network takes a monocular sequence of images as input, so all images share the same camera parameters, that is, ***K**$_j$* = ***K**$_i$*.

4. Preliminaries

In this section, we introduce EpiLoc and EpiSingle in detail. The training pipeline of the Epi-architecture is illustrated in **Fig. 2**. Our method leverages a common depth camera localization architecture in Sec. 4.1. We constrain the network with the absolute pose and the relative pose, which is described in Sec. 4.2. The epipolar geometry constraint is formulated as loss terms in Sec. 4.3. EpiSingle is presented in Sec. 4.4.

4.1 Camera pose regression

The DNN of EpiLoc consists of an encoder block, a localizer block, and a regressor block. The encoder block accepts images as input and extracts features. We employ ResNet34 [11] as the encoder block and obtain the output feature map of the last convolution layer, and then followed by a global average pooling layer. The new feature map will be reshaped and forwarded to the localizer block, a 2048-d FC layer, and a ReLU layer and dropout layer with p = 0.5. The regressor block is two separate 3-d FC layers for regressing translation and rotation, respectively. Instead of using a 4-d FC layer to restore unit quaternion [16],[14],[22],[30], we follow MapNet [3] to restore the logarithm of a unit quaternion (***log q***) [1]. ***g = log q*** is a 3-d vector, which is not over-parameterized. Because the rotation matrix is necessary for the epipolar constrains, we will convert the output ***g*** of EpiLoc to a rotation matrix ***R*** during training.

Given the quaternion ***q***, the logarithm of the quaternion ***g*** and the rotation matrix ***R***, where ***q = (w,v)*** is a unit quaternion, ***w*** is the real part of ***q***, ***v = (x,y,z)*** is its imaginary part, the conversions between them can be calculated by:

From ***q = (w,v)*** to ***g***:

$$g = \begin{cases} \dfrac{v}{\|v\|} \cos^{-1} w & if \;\; \|v\| \neq 0 \\ 0 & otherwise \end{cases} \tag{2}$$

From ***g*** to ***q***:

$$q = (\cos\|g\|, \frac{g}{\|g\|} \sin\|g\|) \tag{3}$$

From ***q=(w,x,y,z)*** to ***R***:

$$R = \begin{bmatrix} 1-2y^2-2z^2 & 2xy+2wz & 2xz-2wy \\ 2xy-2wz & 1-2x^2-2z^2 & 2yz+2wx \\ \text{2xz + 2wy} & \text{2yz - 2wx} & 1-2x^2-2y^2 \end{bmatrix} \tag{4}$$

4.2 Joint learning of pose

In prior works [16],[14],[22],[30],[15],[12],[31],[5],[3],[34], the training loss of rotation is the regression norm, $\boldsymbol{L_1} = \| \cdot \|_1$ or $\boldsymbol{L_2} = \| \cdot \|_2$. Especially, it is noted that the $\boldsymbol{L_1}$ norm performed better [15], but we find that the orientation error between two rotation matrices is more suitable for the epipolar constraint, which can be computed by $\mathrm{AE}(\cdot)$:

$$\mathrm{AE}(R_1, R_2) = \frac{180}{\pi}\cos^{-1}\left(\frac{\mathrm{trace}(R_1 R_2^{-1}) - 1}{2}\right) \tag{5}$$

As our network takes adjacent images as input, EpiLoc minimizes the loss of the absolute pose for each image as well as the loss of relative pose between adjacent images during training, similar to previous works [5],[3],[34]. Given the adjacent images $\boldsymbol{I_i}$ where $\boldsymbol{I = j\,,\,j{+}1}$ and their ground truth poses ($\hat{t}_i$, $\hat{R}_i$) represented by the translation $\hat{t}_i$ and the rotation matrix $\hat{R}_i$, the corresponding estimated pose of EpiLoc is ($\boldsymbol{t_i}$, $\boldsymbol{g_i}$). $\boldsymbol{g_i}$ will be converted to rotation matrix $\boldsymbol{R_i}$. The losses of joint learning of pose are defined as:

$$L_{abs} = \left\| t_i - \hat{t}_i \right\|_1 e^{-\beta} + \beta + \mathrm{AE}(R_i, \hat{R}_i) e^{-\gamma_1} + \gamma_1 \tag{6}$$

$$L_{rel} = \left\| t_r - \hat{t}_r \right\|_1 e^{-\beta} + \beta + \mathrm{AE}(R_r, \hat{R}_r) e^{-\gamma_2} + \gamma_2 \tag{7}$$

where $\boldsymbol{t_r = t_j - t_{j+1}}$ and $\boldsymbol{R_r = R_{j+1} R^{-1}_j}$ are the relative translation and rotation. β, γ_1 and γ_2 are learnable weights used to balance the translation loss and rotation loss.

4.3 Constraint of epipolar geometry

The core of our work is to establish the epipolar geometry constraint through the known camera internal parameters, the pixel correspondence generated by Flownet2 [13], and the camera pose estimated by EpiLoc, to constrain geometric relationships on pixels. To be specific, we should minimize the value of $\boldsymbol{p^T{}_2 F p_1}$ mentioned in Sec. 3.1, However, it is only an algebraic error that does not reflect the real geometric distances. Therefore, we creatively minimize the symmetrical epipolar distance (SED) [10] instead, which is a pixel-level metric and represents the distance from the pixel to its potential epipolar line, as shown in **Fig. 1**. The SED between image $\boldsymbol{I_j}$ and $\boldsymbol{I_{j+1}}$ is defined as:

$$\mathrm{SED}(I_j, I_{j+1}, F) = \sum_i \sigma_i \left(\frac{\left| \overline{q}_i^T F \overline{p}_i \right|}{\sqrt{(F\overline{p}_i)_1^2 + (F\overline{p}_i)_2^2}} + \frac{\left| \overline{q}_i^T F \overline{p}_i \right|}{\sqrt{(F^T\overline{q}_i)_1^2 + (F^T\overline{q}_i)_2^2}} \right) \tag{8}$$

where $\boldsymbol{F}$ is the fundamental matrix between image $\boldsymbol{I_j}$ and $\boldsymbol{I_{j+1}}$ and is computed by (1). $\overline{p}_i$ and $\overline{q}_i$ are homogeneous coordinate representations of pixels $\boldsymbol{p_j \in I_j}$ and $\boldsymbol{q_i \in I_{j+1}}$, and they are the projection pixels of the same 3D point on different images. For instance, $(F\overline{p}_i)_x^2$ represents the square of the x-th entry of the vector $F\overline{p}_i$. $\sigma_i \in \{0,1\}$ is the weight of each pixel and will be set 1 if the SED of ground-truth poses is less than 4. The relative pose rather than the absolute pose will be paid more attention by our network when directly calculating the symmetrical epipolar distance between the estimated poses. In other words, our network will estimate accurate relative pose but is not tailored to absolute pose. To overcome it, we propose the cross symmetrical epipolar distance (CSED) to compute the fundamental matrix $\boldsymbol{F_{mn}}$ between the estimated and ground-truth poses, where $\boldsymbol{m,n \in \{p,t\}}$, and $\boldsymbol{m}$ represent the pose of image $\boldsymbol{I_j}$, $\boldsymbol{n}$ represents the pose of image $\boldsymbol{I_{j+1}}$, $\boldsymbol{p}$ represents the estimated pose of EpiLoc and $\boldsymbol{t}$ represents the ground-truth pose. Note that $\boldsymbol{F_{tt}}$ is used to compute σ_i. The CSED is calculated by:

$$\mathrm{CSED}(I_j, I_{j+1}) = (\mathrm{SED}(I_j, I_{j+1}, F_{pt}) + \mathrm{SED}(I_j, I_{j+1}, F_{tp}) + \mathrm{SED}(I_j, I_{j+1}, F_{pp}))/3 \tag{9}$$

Similar to the loss of pose, the epipolar loss with learnable weight ε is defined as:

$$L_{epi} = \mathrm{CSED}(I_j, I_{j+1})e^{-\varepsilon} + \varepsilon \tag{10}$$

The total loss consisting of the absolute pose loss $\boldsymbol{L_{abs}}$, the relative pose loss $\boldsymbol{L_{rel}}$, and the epipolar geometry loss $\boldsymbol{L_{epi}}$ is defined as:

$$L_{total} = L_{abs} + L_{rel} + L_{epi} \tag{11}$$

4.4 EpiSingle: an extension for non-sequential training data

To adapt the epipolar geometry to the discrete cases, we design EpiSingle to convert the error of a single image pose into the pixel projection error, with the architecture described in **Fig. 2**. The motivation behind this is that the epipolar geometry should map a certain pixel in the one image $\boldsymbol{I}$ (with an estimated pose) to the same location in itself (with a ground-truth pose). We formulate such property as a self-supervised constraint. The symmetrical epipolar distance is defined as:

$$\mathrm{SED}(I, I, F) = \sum_i \frac{\left|\overline{p}_i^T F \overline{p}_i\right|}{\sqrt{(F\overline{p}_i)_1^2 + (F\overline{p}_i)_2^2}} \tag{12}$$

where $\boldsymbol{F}$ is computed by the estimated and ground-truth pose of image $\boldsymbol{I}$, and $\boldsymbol{p_i} \in \boldsymbol{I}$. The self-supervised loss of EpiSingle is defined as:

$$L_{epiS} = \mathrm{SED}(I, I, F)e^{-\varepsilon} + \varepsilon \tag{13}$$

With the self-supervised loss term above, the total loss of EpiSingle is defined as:

$$L_{total} = L_{abs} + L_{epi} \tag{14}$$

5. Experiments

In this section, we show the implementation details, the datasets, and the comparative methods used in experiments. We then validate our proposed approach comparing with other competitors through extensive experiments in Sec. 5.2 and 5.3. Finally, we replace the DNN architecture with RVL [12] to show the generality of our method in Sec. 5.4.

5.1 Implementation details

We adopt PyTorch to implement our approaches on NVIDIA 2080, using Adam solver [17] with weight decay of 0.0005, fixed learning rate of 0.0001, dropout of 0.5, and batch size of 20. The images are cropped to 341x256 and then normalized to have pixel intensities within [-1,1]. During the training on the RobotCar [20], we randomly color-disturbed the image to perform data augmentation, setting the brightness, contrast, and saturation to 0.7 and hue to 0.5. The network of EpiLoc is initialized by the pretrained MapNet [3] for reducing the training time. $\beta, \gamma_1, \gamma_2$ and ε are set to 0, 2, -3, 2. The optical flow between adjacent frames is estimated by Flownet2 [13] before training. All images are scaled to 256x256, and the camera's internal parameters are scaled accordingly.

5.1.1 Datasets

The 7Scenes dataset [25] is an RGB-D database captured at 640x480 resolution and consists of seven different small indoor office scenes, each less than 4 meters in the spatial extent. Each scene contains several sequences. The training and test sets of each scene are also provided. The corresponding ground truth camera poses were obtained by the KinectFusion system. The camera internal parameters (principal point (320,240), focal length (585,585)) were used in the KinectFusion pipeline.

The Oxford RobotCar dataset [20] is a large-scale outdoor scene located in Oxford. The dataset contains complex variations, including weather, lighting, and dynamic objects, which makes RobotCar a difficult challenge in the relocalization task. We use the intermediate image at a resolution of 1280x960 captured by the stereo camera, and the corresponding ground truth poses are obtained by the inertial navigation system (INS). The camera's internal parameters are calculated by the camera model provided by the dataset. We follow MapNet [3] and use two subsets (LOOP and FULL) of RobotCar.

5.1.2 Comparative methods

We compare EpiSingle with PoseNet [14],[15],[16] on 7Scenes [25]. To evaluate the performance of EpiLoc, we compare it to PoseNet15 [16], PoseNet16 [14], LSTM-Pose [30], Hourglass [22], PoseNet-17 [15], AtLoc [31], DSO [7], VidLoc [5], MapNet [3], LSG [34], PoseGan [40], Direct [41], AtLoc+ [31] on the indoor 7Scenes dataset [25], we use PoseNet [14],[15],[16], MapNet [3], LSG [34] and AtLoc [31] as the competing methods on the outdoor RobotCar dataset [20].

5.2 Experiments on 7Scenes

5.2.1 The performance of EpiSingle

For a fair comparison, we compare EpiSingle with PoseNet [14],[15],[16] in **Table 1**. The better performances of EpiSingle prove the effectiveness of pixel-level constraints. In particular, EpiSingle achieves a considerable performance gain in highly texture-repetitive (such as stairs) scenarios. EpiSingle narrows the median translation and rotation errors to 0.21m and 7.89 respectively, outperforming PoseNet [14],[15],[16] by a large margin.

Table 1. The performance comparison of EpiSingle on 7Scenes.
For each scene, we report the median translation and rotation errors of PoseNet [14],[15],[16] and our EpiSingle on 7Scenes. The best results are highlighted.

Methods	**Chess**	**Fire**	**Heads**	**Office**	**Pumpkin**	**Kitchen**	**Stairs**	**Average**
PoseNet [14],[15],[16]	0.11m,4.29°	0.27m,12.2°	0.19m,**12.1°**	0.19m,6.36°	0.22m,5.06°	0.25m,**5.27°**	0.30m,11.3°	0.22m,8.08°
EpiSingle (Ours)	**0.11m,4.07°**	**0.27m,11.2°**	**0.17m**,13.3°	**0.19m,6.25°**	**0.22m,4.53°**	**0.25m**,5.35°	**0.27m,10.5°**	**0.21m,7.89°**

5.2.2 Comparison of EpiLoc and prior methods on 7Scenes

Fig. 3 reports the cumulative distributions for each scene, which proves the better performance of our method.

Table 2 shows the performance comparison between the single-image methods and our EpiLoc. Obviously, our method outperforms these baseline methods, because the lack of geometric information of these methods will bring high uncertainty.

The localization results of the methods based on multiple images and our EpiLoc are summarized in **Table 3**. Our method is not specific to any particular environment, achieving a considerable performance gain compared to MapNet [3], and slightly better than the state-of-the-art method. MapNet [3] proposes a geometry-aware learning paradigm. LSG [34] introduces the convolutional LSTM [33] for an extra VO estimation and uses a soft attention mechanism to augment the features map. AtLoc+ [31] incorporates a self-attention mechanism and temporal constraint between image pairs. These are proved to be valid, but they use the coarse motion-geometry constraint, different from our EpiLoc which establishes the geometric relationship between pixels. A finer-grained geometric constraint helps our network to regress accurate and robust poses.

Table 2. Camera localization results of EpiLoc and single-image methods on 7Scenes. For each scene, we compute the median translation and rotation errors of various single-image methods and our EpiLoc. The best results are highlighted.

Methods	**Chess**	**Fire**	**Heads**	**Office**	**Pumpkin**	**Kitchen**	**Stairs**	**Average**
PoseNet15 [16]	0.32m,6.60°	0.47m,14.0°	0.30m,12,2°	0.48m,7.24°	0.49m,8.12°	0.58m,8.34°	0.48m,13.1°	0.45m,9.94°
PoseNet16 [14]	0.37m,7.24°	0.43m,13.7°	0.31m,12.0°	0.48m,8.04°	0.61m,7.08°	0.58m,7.54°	0.48m,13.1°	0.47m,9.81°
LSTM [30]	0.24m,5.77°	0.34m,11.9°	0.21m,13.7°	0.30m,8.08°	0.33m,7.00°	0.37m,8.83°	0.40m,13.7°	0.31m,9.85°
Hourglass [22]	0.15m,6.17°	0.27m,10.8°	0.19m,11.6°	0.21m,8.48°	0.25m,7.01°	0.27m,10.2°	0.29m,12.5°	0.23m,9.53°
PoseNet17 [15]	0.13m,4.48°	0.27m,11.3°	0.17m,13.0°	0.19m,5.55°	0.26m,4.75°	0.23m,5.35°	0.35m,12.4°	0.23m,8.12°
PoseGan [40]	0.09m,4.58°	0.24m,9.46°	0.17m,13.4°	0.19m,8.80°	**0.16m**,6.28°	0.26m,8.23°	0.28m,10.1°	0.20m,8.70°
AtLoc [31]	0.10m,4.07°	0.25m,11.4°	0.16m,**11.8°**	0.17m,5.34°	0.21m,4.37°	0.23m,5.42°	0.26m,10.5°	0.20m,7.56°
Direct [41]	0.10m,3.52°	0.27m,**8.66°**	0.17m,13.1°	**0.16m**,5.96°	0.19m,3.85°	**0.22m**,5.13°	0.32m,10.6°	0.20m,7.26°
EpiLoc (Ours)	**0.07m,2.71°**	**0.24m,**9.18°	**0.14m,**12.6°	0.18m,**4.45°**	0.18m,**3.22°**	0.23m,**4.60°**	**0.24m,11.0°**	**0.18m,6.82°**

Table 3. Camera localization results of EpiLoc and multi-image methods on 7Scenes. For each scene, we compute the median translation and rotation errors of various multi-image methods and EpiLoc. The best results are highlighted.

Methods	**Chess**	**Fire**	**Heads**	**Office**	**Pumpkin**	**Kitchen**	**Stairs**	**Average**
DSO [7]	0.17m,8.13°	**0.19m**,65.0°	0.61m,68.2°	1.51m,16.8°	0.61m,15.8°	0.23m,10.9°	0.26m,21.3°	0.26m,29.4°
VidLoc [5]	0.18m,NA	0.26m,NA	0.14m,NA	0.26m,NA	0.36m,NA	0.31m,NA	0.26m,NA	0.25m,NA
MapNet [3]	0.08m,3.25°	0.34m,11.9°	0.18m,13.3°	**0.17m**,5.15°	0.22m,4.02°	0.23m,4.93°	0.30m,12.1°	0.21m,7.77°
LSG [34]	0.09m,3.28°	0.27m,10.8°	0.17m,12.7°	0.18m,5.45°	0.20m,3.69°	0.23m,4.92°	**0.23m**,11.3°	0.19m,7.47°
AtLoc+ [31]	0.10m,3.18°	0.26m,10.8°	0.14m,**11.4°**	**0.17m**,5.16°	0.20m,3.94°	**0.16m**,4.90°	0.29m,**10.2°**	0.19m,7.08°
EpiLoc (Ours)	**0.07m**,**2.71°**	0.24m,**9.18°**	**0.14m**,12.6°	0.18m,**4.45°**	**0.18m**,**3.22°**	0.23m,**4.60°**	0.24m,11.0°	**0.18m**,**6.82°**

5.3 Experiments on Oxford RobotCar

5.2.1 The quantitative comparison

Table 4. Camera localization results on RobotCar. For each scene, we compute the mean translation / rotation errors of various methods and our EpiLoc. The best results are highlighted.

Methods	**LOOP1**	**LOOP2**	**FULL1**	**FULL2**	**Average**
PoseNet [14],[15],[16]	10.6m,4.46°	11.4m,5.08°	42.0m,11.0°	60.1m,12.8°	31.0m,8.32°
MapNet [3]	9.29m,4.34°	8.89m,4.07°	32.5m,8.61°	50.6m,10.8°	25.3m,6.95°
LSG [34]	9.07m,3.31°	9.19m,3.53°	31.7m,4.51°	53.5m,**8.60°**	25.8m,4.99°
AtLoc [31]	8.61m,4.58°	8.86m,4.67°	29.6m,12.4°	48.2m,11.1°	23.8m,8.19°
EpiLoc (Ours)	**8.09m,3.03°**	**8.48m,3.03°**	**16.2m,2.95°**	**38.2m**,9.13°	**17.8m,4.53°**

Table 4 shows the results of our EpiLoc and other competitive methods on RobotCar. Compared to the state-of-the-art AtLoc [31], EpiLoc reduces the translation error from 23.8m to 17.8m and the rotation error from 8.19 ° to 4.53 ° . The pixel-level epipolar geometry constraint helps our method achieve a considerable performance gain.

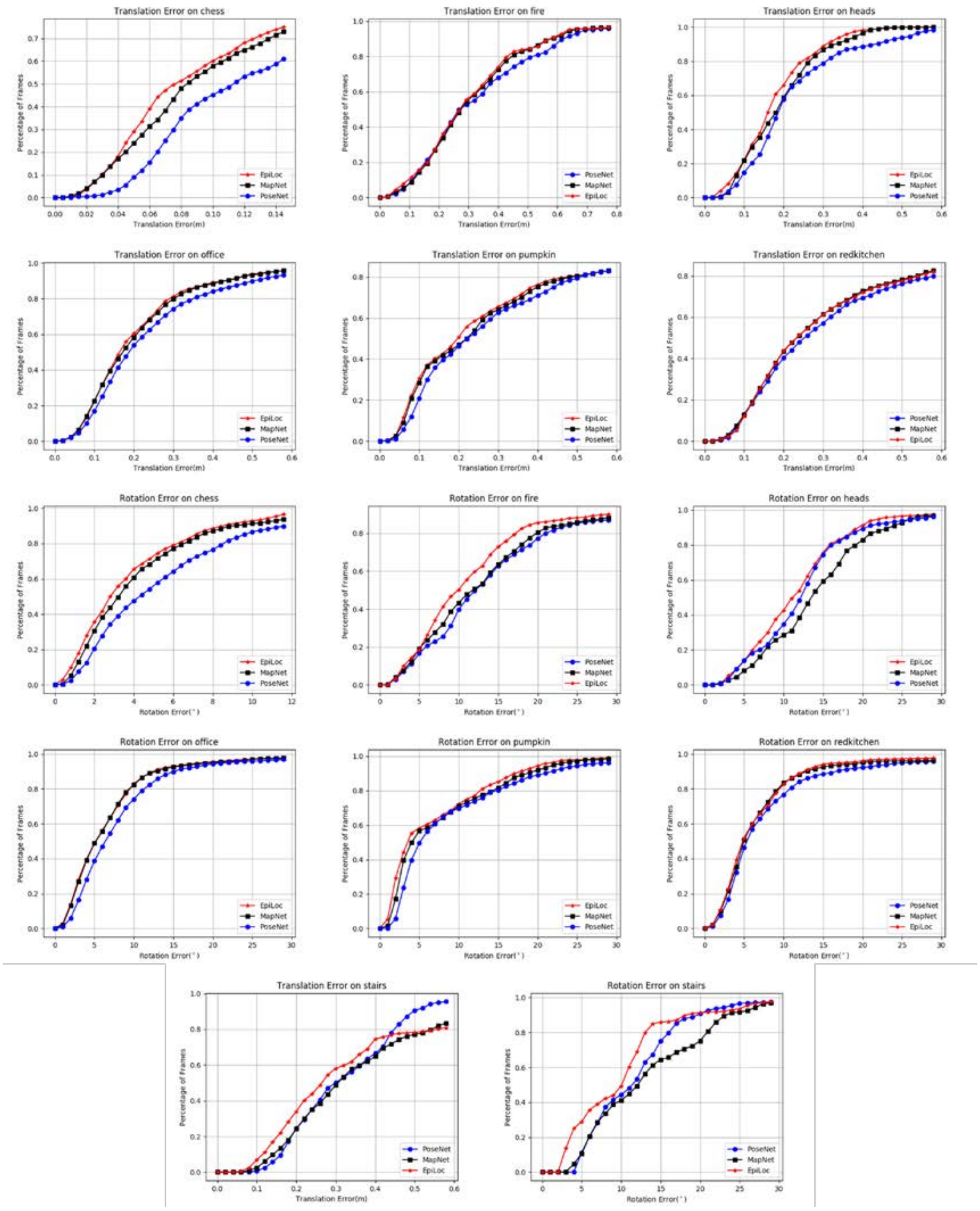
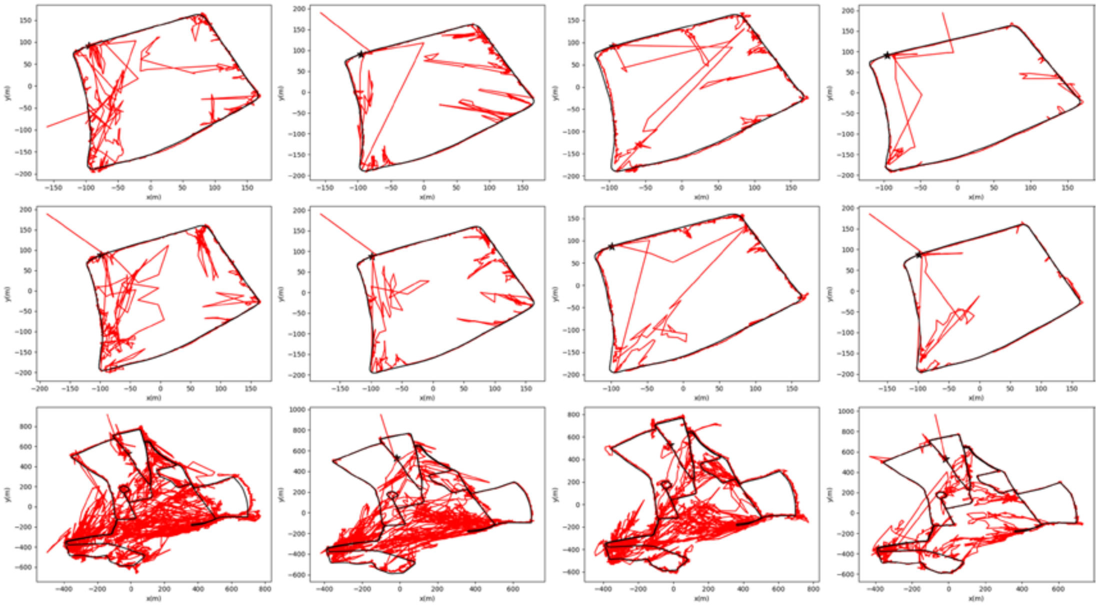


Fig. 3. The cumulative distributions of translation and rotation for each scene on 7Scenes. The first two rows are the translation error on chess, fire, heads, office, pumpkin, redkitchen respectively. The third and fourth rows are the rotation errors of the corresponding scene. The last row is the translation (left) and rotation (right) errors on stairs. The x-axis is the translation or rotation error and the y-axis is the accuracy, *i.e.* the percentage of frames whose localization error is less than the value.

5.2.1 The qualitative comparison

Fig. 4 shows the predicted trajectory of the corresponding methods. **Fig. 5** shows the cumulative distribution translation and rotation error of Pose-Net [14],[15],[16], MapNet [3], AtLoc [31], and our EpiLoc on LOOP1 and FULL1 scenes. PoseNet [14],[15],[16] predicts the pose of images through a single image, which generates a large error. MapNet [3] achieves a more accurate pose through the motion constraint between image pairs. Self-attention is introduced to the AtLoc [31], forcing the network to focus on more geometrically robust objects and features. Our EpiLoc establishes a finer-grained geometric constraint and produces fewer outliers in outdoor LOOP and FULL scenes.



(a) PoseNet [14],[15],[16] (b) MapNet [3] (c) AtLoc [31] (d) Ours

Fig. 4. The predicted trajectory on the LOOP1 (top), LOOP2 (middle) and FULL1 (bottom) of the RobotCar dataset.
The red and black lines are the predicted and ground truth trajectory respectively. The black star represents the first frame.

5.4 Generality study

Our EpiLoc can also be applied to other networks to achieve better performance. RVL [12] proposes a prior guided dropout module and a composite self-attention module, which can guide the networks to ignore the dynamic objects. This helps RVL [12] achieve excellent results on the RobotCar dataset [20], which is a vehicle-mounted dataset with a large number of dynamic objects such as pedestrians, vehicles, *etc*. In this section, in order to further verify the validity of the pixel-level geometric constraint, we replace our vanilla DNN with the networks of RVL [12] and report their errors (**Table 5**) and cumulative distributions (**Fig. 6**) on the LOOP scene. Note that the split of the train/test sequences in RVL [12] is inconsistent with ours, which is the reason we did not directly compare it in the front.

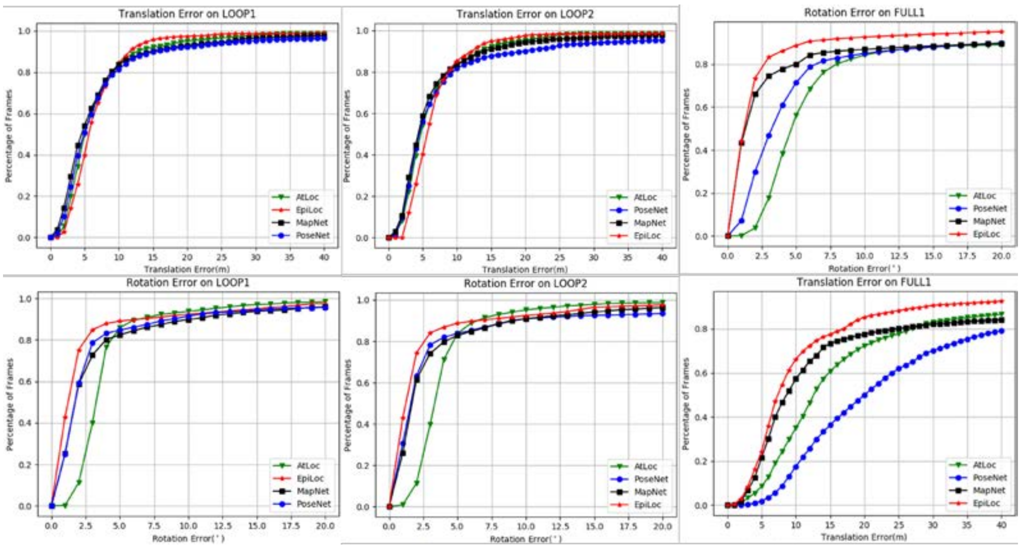


Fig. 5. The cumulative distributions on RobotCar.
We report the translation and rotation errors for LOOP1 (right), LOOP2 (middle) and FULL1 (left) scenes of the RobotCar dataset. The x-axis is the translation or rotation error and the y-axis is the accuracy, *i.e.* the percentage of frames whose localization error is less than the value.

Table 5. Camera localization results on LOOP.
We compute the mean translation / rotation errors of RVL [12] with/without EpiLoc. The best results are highlighted. Note that we follow the split of train/test in RVL [12] to create the LOOP.

Methods	Median	Mean
RVL [12]	4.11m,1.56°	6.33m,2.70°
RVL-EpiLoc	**3.22m,1.14°**	**5.53m,2.21°**

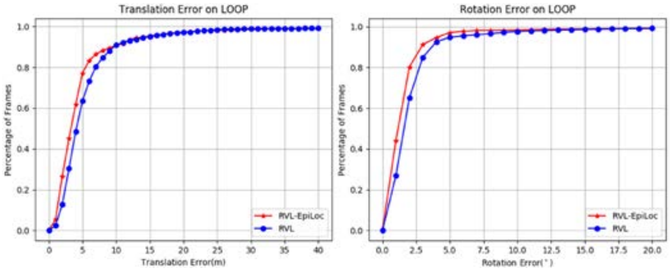


Fig. 6. The cumulative distributions of RVL [12] and RVL-EpiLoc.

6. Conclusions

Camera localization is a fundamental task in computer vision, which is widely used in many fields, such as augmented reality, robots and so on. We propose EpiLoc, which imposes a finer-grained geometric constraint on the network through the epipolar geometry and optical flow during training, and it achieves a considerable improvement. We also design EpiSingle for non-sequential images without access to the corresponding relationship of pixels, which further boosts the applicability of the epipolar geometry. Our proposed method can also be combined with other methods to achieve better results by replacing the network. Extensive experiments on both the 7Scenes and RobotCar datasets show the effectiveness of epipolar geometric constraints.

Acknowledgments

This work was supported by the National Key Research and Development Plan of China [Grant numbers. No, 2020YFC2006000] and the National Natural Science Foundation of China (NSFC) [Grant numbers. No.72071089].

References

[1] Altmann, Simon L, *Rotations, quaternions, and double groups*, Courier Corporation, 2005.

[2] Badrinarayanan, Vijay, Alex Kendall, and Roberto Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, 39(12), 2481-2495, 2017. Article (CrossRef Link)

[3] Brahmbhatt, Samarth, et al, "Geometry-aware learning of maps for camera localization," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. Article (CrossRef Link)

[4] Chen, Yuhua, Cordelia Schmid, and Cristian Sminchisescu, "Self-supervised learning with geometric constraints in monocular video: Connecting flow, depth, and camera," in *Proc. of the IEEE/CVF International Conference on Computer Vision*, 2019. Article (CrossRef Link)

[5] Clark, Ronald, et al, "Vidloc: A deep spatio-temporal model for 6-dof video-clip relocalization," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. Article (CrossRef Link)

[6] Dosovitskiy, Alexey, et al, "Flownet: Learning optical flow with convolutional networks," in *Proc. of the IEEE international conference on computer vision*, 2015. Article (CrossRef Link)

[7] Engel, Jakob, Vladlen Koltun, and Daniel Cremers, "Direct sparse odometry," *IEEE transactions on pattern analysis and machine intelligence*, 40(3), 611-625, 2018. Article (CrossRef Link)

[8] Fischler, Martin A., and Robert C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, 24(6), 381-395, 1981. Article (CrossRef Link)

[9] Godard, Clément, Oisin Mac Aodha, and Gabriel J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proc. of the IEEE conference on computer vision and pattern recognition*, 2017. Article (CrossRef Link)

[10] Andrew, Alex M, "Multiple view geometry in computer vision," *Kybernetes*, vol. 30, no. 9/10, pp. 1333-1341, 2001. Article (CrossRef Link)

[11] He, Kaiming, et al, "Deep residual learning for image recognition," in *Proc. of the IEEE conference on computer vision and pattern recognition*, pp. 770-778, 2016. Article (CrossRef Link)

[12] Huang, Zhaoyang, et al, "Prior guided dropout for robust visual localization in dynamic environments," in *Proc. of the IEEE/CVF International Conference on Computer Vision*, 2019. Article (CrossRef Link)

[13] Ilg, Eddy, et al, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *Proc. of the IEEE conference on computer vision and pattern recognition*, 2017. Article (CrossRef Link)

[14] Kendall, Alex, and Roberto Cipolla, "Modelling uncertainty in deep learning for camera relocalization," in *Proc. of 2016 IEEE international conference on Robotics and Automation (ICRA)*, IEEE, 2016. Article (CrossRef Link)

[15] Kendall, Alex, and Roberto Cipolla, "Geometric loss functions for camera pose regression with deep learning," in *Proc. of the IEEE conference on computer vision and pattern recognition*, 2017. Article (CrossRef Link)

[16] Kendall, Alex, Matthew Grimes, and Roberto Cipolla, "Posenet: A convolutional network for real-time 6-dof camera relocalization," in *Proc. of the IEEE international conference on computer vision*, 2015. Article (CrossRef Link)

[17] Kingma, Diederik P., and Jimmy Ba, "Adam: A method for stochastic optimization," in *Proc. of International Conference on Learning Representations*, 2015. Article (CrossRef Link)

[18] Kocabas, Muhammed, Salih Karagoz, and Emre Akbas, "Self-supervised learning of 3d human pose using multi-view geometry," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. Article (CrossRef Link)

[19] Li, Yunpeng, Noah Snavely, and Daniel P. Huttenlocher, "Location recognition using prioritized feature matching," in *Proc. of European conference on computer vision*, pp. 791-804, 2010. Article (CrossRef Link)

[20] Maddern, Will, et al, "1 year, 1000 km: The oxford robotcar dataset," *The International Journal of Robotics Research*, 36(1), 3-15, 2017. Article (CrossRef Link)

[21] Mahajan, Dhruv, et al, “Exploring the limits of weakly supervised pretraining,” in *Proc. of the European conference on computer vision (ECCV)*, pp. 185-201, 2018. Article (CrossRef Link)

[22] Melekhov, Iaroslav, et al, “Image-based localization using hourglass networks,” in *Proc. of the IEEE international conference on computer vision workshops*, 2017. Article (CrossRef Link)

[23] Sattler, Torsten, et al, “Hyperpoints and fine vocabularies for large-scale location recognition,” in *Proc. of the IEEE International Conference on Computer Vision*, 2015. Article (CrossRef Link)

[24] Sattler, Torsten, et al, “Large-scale location recognition and the geometric burstiness problem,” in *Proc. of the IEEE conference on computer vision and pattern recognition*, 2016. Article (CrossRef Link)

[25] Shotton, Jamie, et al, “Scene coordinate regression forests for camera relocalization in RGB-D images,” in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013. Article (CrossRef Link)

[26] Svarm, Linus, et al, “Accurate localization and pose estimation for large 3d models,” in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014. Article (CrossRef Link)

[27] Szegedy, Christian, et al, “Going deeper with convolutions,” in *Proc. of the IEEE conference on computer vision and pattern recognition*, 2015. Article (CrossRef Link)

[28] Tan, Mingxing, and Quoc Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” in *Proc. of International Conference on Machine Learning*, 2019. Article (CrossRef Link)

[29] Tobin, Joshua, Wojciech Zaremba, and Pieter Abbeel, “Geometry-aware neural rendering,” *Advances*,” in *Proc. of 33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, 2019. Article (CrossRef Link)

[30] Walch, Florian, et al, “Image-based localization using lstms for structured feature correlation,” in *Proc. of the IEEE International Conference on Computer Vision*, 2017. Article (CrossRef Link)

[31] Wang, Bing, et al, “Atloc: Attention guided camera localization,” in *Proc. of the AAAI Conference on Artificial Intelligence*, 34(06), 10393-10401, 2020. Article (CrossRef Link)

[32] Westlake, Nicholas, Hongping Cai, and Peter Hall, “Detecting people in artwork with cnns,” in *Proc. of European Conference on Computer Vision*, Springer, Cham, pp. 825-841, 2016. Article (CrossRef Link)

[33] Xingjian, S. H. I., et al, “Convolutional LSTM network: A machine learning approach for precipitation nowcasting,” *Advances in neural information processing systems*, 2015. Article (CrossRef Link)

[34] Xue, Fei, et al, “Local supports global: Deep camera relocalization with sequence enhancement,” in *Proc. of the IEEE/CVF International Conference on Computer Vision*, 2019. Article (CrossRef Link)

[35] Yao, Yuan, Yasamin Jafarian, and Hyun Soo Park, “Monet: Multiview semi-supervised keypoint detection via epipolar divergence,” in *Proc. of the IEEE/CVF International Conference on Computer Vision*, 2019. Article (CrossRef Link)

[36] Zhao, Shanshan, et al, “Geometry-aware symmetric domain adaptation for monocular depth estimation,” in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. Article (CrossRef Link)

[37] Zhong, Yiran, et al, “Unsupervised deep epipolar flow for stationary or dynamic scenes,” in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. Article (CrossRef Link)

[38] Zhou, Tinghui, et al, “Unsupervised learning of depth and ego-motion from video,” in *Proc. of the IEEE conference on computer vision and pattern recognition*, 2017. Article (CrossRef Link)

[39] Zhu, Yi, et al, “Improving semantic segmentation via video propagation and label relaxation,” in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. Article (CrossRef Link)

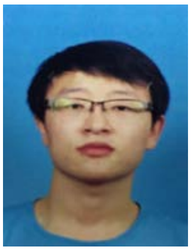
[40] Liu, Kanglin, Qing Li, and Guoping Qiu, “PoseGAN: A pose-to-image translation framework for camera localization,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 166, pp. 308-315, 2020. Article (CrossRef Link)

[41] Chen, Shuai, Zirui Wang, and Victor Prisacariu, "Direct-PoseNet: Absolute Pose Regression with Photometric Consistency," in *Proc. of International Conference on 3D Vision (3DV)*, 2021. Article (CrossRef Link)

Luoyuan Xu is currently working toward the Ph.D. degree with the Huazhong University of Science and Technology, Wuhan, China. His major research interests include computer vision and multiview stereo. Xu received the B.S. degree from the Wuhan University of Technology, Wuhan, China, in 2018.

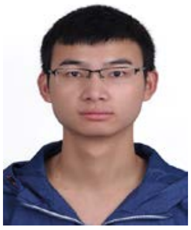
Tao Guan received the Ph.D. degree from the Huazhong University of Science and Technology, Wuhan, China in 2008. He is currently a professor in the School of Computer Science and Technology, Huazhong University of Science and Technology. His research interests include mobile visual search and mobile augmented reality.

Yawei Luo is currently a postdoctoral researcher with CCAI, College of Computer Science and Technology, Zhejiang Uni- versity, Hangzhou, China. From 2017 to 2019, he was a visiting Ph.D. student with ReLER Lab, AAII, University of Technology Sydney, Ultimo, NSW, Australia. His research interests include semantic segmentation, domain adaptation, and 3D recon- struction. Luo received the B.S. and Ph.D. degrees from the Huazhong University of Science and Technology, Wuhan, China, in 2013 and 2020, respectively.



Yuesong Wang is currently a postdoctoral researcher with the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China. His research interests include computer vision and deep learn- ing. Wang received the Ph.D. degree in computer science from the Huazhong University of Science and Technology, in 2021.

Zhuo Chen received the B.S. degree and is currently pursuing a Phd degree at Huazhong University of Science and Technology, Hubei, China. His major research interests include computer graphics and computer vision.



Wenkai Liu is currently working toward the Ph.D. degree with the Huazhong University of Science and Technology, Hubei, China. His major research interests include computer graphics and computer vision. Liu received the B.S. degree from Hunan University, Changsha, China, in 2018.