Local Supports Global: Deep Camera Relocalization with Sequence Enhancement

Fei Xue[1,2], Xin Wang[2,3], Zike Yan[2,3], Qiuyuan Wang[2,3], Junqiu Wang[4], and Hongbin Zha[2,3]

[1]UISEE Technology Inc.
[2]Key Laboratory of Machine Perception, Peking University
[3]PKU-SenseTime Machine Vision Joint Lab, Peking University
[4]Beijing Changcheng Aviation Measurement and Control Institute, AVIC
{feixue, xinwang_cis, zike.yan, wangqiuyuan}@pku.edu.cn
jerywangjq@foxmail.com, zha@cis.pku.edu.cn

Abstract

We propose to leverage the local information in image sequences to support global camera relocalization. In contrast to previous methods that regress global poses from single images, we exploit the spatial-temporal consistency in sequential images to alleviate uncertainty due to visual ambiguities by incorporating a visual odometry (VO) component. Specifically, we introduce two effective steps called content-augmented pose estimation and motion-based refinement. The content-augmentation step focuses on alleviating the uncertainty of pose estimation by augmenting the observation based on the co-visibility in local maps built by the VO stream. Besides, the motion-based refinement is formulated as a pose graph, where the camera poses are further optimized by adopting relative poses provided by the VO component as additional motion constraints. Thus, the global consistency can be guaranteed. Experiments on the public indoor 7-Scenes and outdoor Oxford RobotCar benchmark datasets demonstrate that benefited from local information inherent in the sequence, our approach outperforms state-of-the-art methods, especially in some challenging cases, e.g., insufficient texture, highly repetitive textures, similar appearances, and over-exposure.

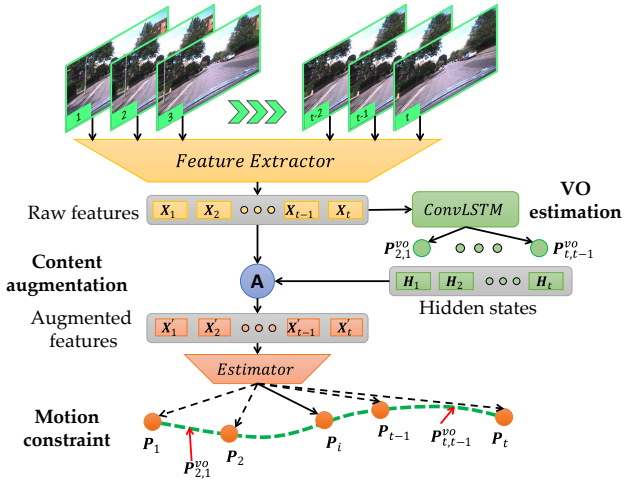


Figure 1: Overview of our framework. Sequential images are first encoded as high-level features which are then fed into the convolutional LSTM [28] for VO estimation. These features are augmented individually by searching co-visibility from hidden states in recurrent units before they are fed into the estimator for global pose regression. Predicted poses are optimized within a pose graph in which the predicted global poses and relative poses represent the nodes and edges, respectively.

1. Introduction

Visual relocalization is a fundamental task in computer vision. With various applications in robotics, autonomous driving, and virtual/augmented reality, visual relocalization has been studied for decades and a considerable amount of geometry-based systems have been developed [18, 23, 24, 31, 32]. Recently, with the success of deep learning techniques in computer vision tasks, global poses can be predicted by exploiting Convolutional Neural Networks (CNNs) [4, 14, 15, 20, 26] and Recurrent Neural Networks (RNNs) [7, 34] in an end-to-end fashion. Most of these methods recover camera poses from single images. However, it is not easy to estimate stable poses from single images due to insufficient information or visual ambiguities

2380-7504/19/$31.00 ©2019 IEEE
DOI 10.1109/ICCV.2019.00293

such as repetitive textures, similar appearances, and the accuracy may also degrade with noise and over-exposure. As a sequence provides locally consistent spatio-temporal cues over consecutive frames, they have potentials to overcome limitations that single images based relocalization suffers.

In order to best utilize the local consistency, we delve into the nature of sequential data in terms of content and motion. First, in contrast to single images, a sequence can build a local map aggregating contents from consecutive frames from different viewpoints [27]. Thus, a sequence provides wider field of view and can distinguish stably detected features. Ambiguities arisen from single images can be alleviated with the local map. Additionally, considering the continuous nature of camera motions, relative poses can be exploited as additional constraints to mitigate the uncertainty that global pose estimation undergoes [8, 21].

In this paper, we propose to learn camera relocalization from sequences by incorporating a deep VO component. The VO sub-network accepts sequential images as input and predicts relative poses between consecutive frames. The hidden states that preserve historical information of sequence are taken as local maps. Specifically, our model learns camera relocalization by employing a two-step strategy. Raw features are first augmented according to the covisibility in the local map for global pose prediction, namely content-augmentation module; VO estimation is then taken as a local motion constraint to enforce the consistency of predicted global camera poses within a pose graph.

The overview of our architecture is shown in Fig. 1. The encoder extracts features from images. The VO sub-network takes sequential features as input and calculates relative poses using convolutional LSTMs (Long Short-Term Memory) [28]. The extracted features are then augmented using hidden states with a soft attention, and fed into a pose predictor to regress camera poses for each view respectively. Finally, with vertexes denoting global poses and edges representing relative poses, a pose graph is established to enforce the consistency of predicted results. Our model learns camera relocalization and VO estimation jointly in a unified model, and can be used in an end-to-end fashion. Our contributions can be summarized as follows:

- We propose to take advantages of the local information in a sequence to support global camera poses estimation by incorporating a VO component.
- Instead of raw features, our model estimates global poses from contents augmented based on covisibility in local maps.
- We adopt VO results as additional motion constraints to refine global poses within a pose graph during training and testing processes.

Our approach outperforms state-of-the-art methods in both the indoor 7-Scenes and outdoor Oxford RobotCar datasets, especially in challenging scenarios. The rest of this paper is organized as follows. In Sec. 2, related works on relocalization and visual odometry are introduced. In Sec. 3, our framework is described in detail. The performance of our approach is compared with previous methods in Sec. 4. Finally, we conclude the paper in Sec. 5.

2. Related work

We first introduce relevant methods for camera relocalization, and then give a brief discussion on related VO algorithms.

Visual Relocalization Image-based relocalization algorithms recover the global pose approximately using the one of the most similar image in the database [1, 13]. The approximation-like nature results in low accuracy. Structure-based algorithms require a scene 3D representation and search correspondences between 3D points and extracted features from a query image to establish 2D-3D matches [2, 3, 31]. Then, the camera poses are estimated by applying RANSAC and solving a Perspective-n-Point problem [6, 15, 24]. Despite their promising performance, they require high computational cost, accurate intrinsic calibration and initialization for projection.

Recent works show that global camera poses can be estimated using a neural network in an end-to-end fashion. PoseNet [16] is the first to regress 6-DoF camera poses directly with deep neural networks in an end-to-end fashion. A series of following works extend PoseNet by modeling the uncertainty of poses with Bayesian CNNs [14], introducing the geometric loss [15], and correcting the structured features using LSTMs [34]. Melekhov *et al*. [20] add skip connections on ResNet34 [11] for preserving the fine-grained information. All abovementioned methods estimate camera localization from single images.

VidLoc [7] and MapNet [4] are closely related to our work. VidLoc [7] accepts video clips as input and adopts regular bidirectional LSTMs to model the sequence. Although LSTMs can partially enhance observations, it cannot remember historical knowledge for a long time [30], resulting in poor performance in processing long sequences. We instead employ a soft attention mechanism to remedy the finite capacity of recurrent units. Moreover, additional motion constraints are exploited to enforce the consistency of poses, which is ignored in VidLoc [7]. MapNet [4] enforces the motion consistency between predicted global poses during training. However, the input for each view is only a single image, and it requires extra data obtained from GPS or VO/SLAM systems [8, 21] for pose refinement. In contrast, our model takes both the contents and motion cues from a sequence into consideration. Besides, the motion constraints can be performed in both training and testing

stages, as our model directly provides relative poses from the VO component.

Visual Odometry Traditionally, visual odometry is solved by minimizing reprojection [21] or photometric errors [8, 9]. Recently, a number of learning-based VO systems have been developed. Some leverage CNNs to predict ego-motions from videos [38, 40] or relative poses between paired images [33, 39]. VO can also be formulated as sequence-to-sequence problem and solved by a LSTM for sequence modeling [35–37]. Benefited from the local temporal information preserved in LSTMs, relative results are predicted with promising accuracy.

In this paper, we embed a deep VO component to assist relocalization. The hidden states with temporal information are taken as local maps to augment the raw features of each view. Moreover, the relative poses obtained from the VO component are exploited as motion constraints to enforce the consistency of global poses within a pose graph.

3. Method

In this section, we introduce our deep camera relocalization framework in detail. An overall network is illustrated in Fig. 2. The feature extractor module encodes images into high-level features for VO estimation in Sec. 3.1. The extracted contents are augmented for global poses prediction in Sec. 3.2. Estimated global poses are finally optimized along with VO results within a pose graph in Sec. 3.3.

3.1. Feature Extraction and VO Estimation

Feature Extraction Similar to [4, 20], we adopt the modified ResNet34 [11] to extract features from images. The last global average pooling (GAP) layer and fully-connected (FC) layer are discarded, producing 3D-tensors as feature maps. As shown in Fig. 2, since camera relocalization and visual odometry are two closely related subtasks, the similarity can also be inherited in the feature space by sharing weights of feature extraction module. To be specific, the output of our modified ResNet34 for frame t is taken as raw feature X_t. Two consecutive raw features are concatenated along the channel dimension and passed through a group of residual networks to obtain fused features X_t^{vo} for the VO task. This group of residual networks shares the same number of *BasicBlock* with the last residual layer in ResNet34. It is worthy to note that the number of input channels is changed and the pooling operation is disabled.

Visual Odometry Estimation Learning visual odometry from image sequences using LSTMs [12] for temporal reasoning has been proved effective [35, 36]. The regular LSTM [12] cannot retain the spatial connections of features with only 1D vectors as input. We adopt the convolutional LSTM [28] as our recurrent unit following the work in [36].

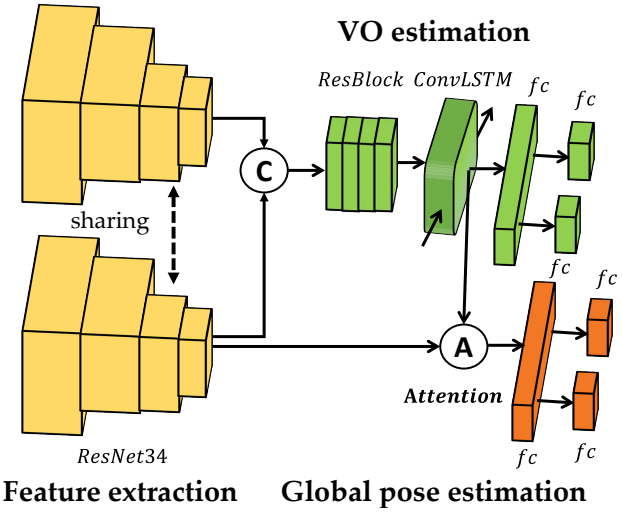


Figure 2: ResNet34 [11] with the last two layers discarded is used to extract features with 512 channels. Features of two consecutive frames are concatenated and passed through the *ResBlock* consisting of 3 *BasicBlock*s for fusion. Fused features are then fed into a convolutional LSTM [28] for VO estimation. Raw features are augmented based on the co-visibility in local maps using a soft attention mechanism. Outputs of recurrent units and augmented features are vectorized to calculate the relative and global 6-DoF poses via 3 FC layers, respectively. The output channels of the first FC layer for both global and relative poses estimation are 2048, while the left are 3.

The fused feature X_t^{vo} is fed into the recurrent unit as:

$$H_t, O_t = \mathcal{U}(X_t^{vo}, H_{t-1}) \,, \tag{1}$$

where H_{t-1} and H_t represent hidden states at time $t-1$ and t, respectively. O_t denotes the output 3D tensor of the recurrent unit $\mathcal{U}$ at time t. It is passed through a GAP layer and three FC layers (with output channels of 2048, 3 and 3) to calculate the 6-DoF relative pose $P_{t,t-1}^{vo}$.

Intuitively, the feature flow passing through recurrent units is filtered and fused via the *gates* in convolutional LSTMs. Hidden states with historical knowledge preserved are viewed as local maps to boost the relocalization task.

3.2. Content-augmented Pose Estimation

We aim at utilizing multiple observations from a sequence to reduce the ambiguity of a single image. Since images are high-dimension data with much redundant information, learning information from raw data is ineffective [35]. On the other hand, LSTMs cannot preserve long-term knowledge [30]. Hence, fusing features instead of images using bidirectional LSTMs is incapable of processing

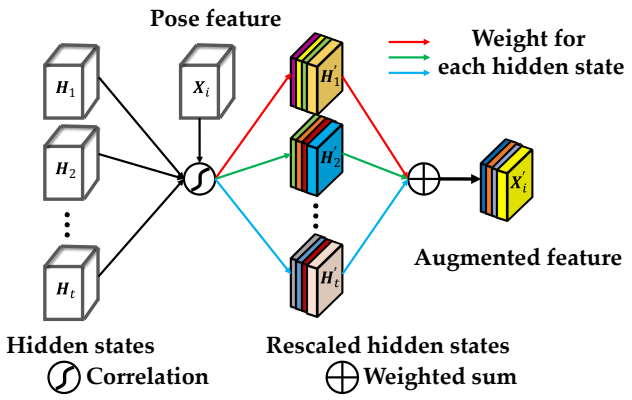


Figure 3: We search the co-visible contents from all hidden states with the guidance of raw features. The selection is a soft attention by re-scaling each hidden state in temporal domain and each channel of hidden states in spatial domain, respectively.

long sequences. To deal with the problem, we take inspirations from classic SLAM systems [21] by searching the co-visibility relations from local maps built in the VO stream.

Content Augmentation In our work, both observations and local maps are represented implicitly as features. Hence, the correlation in the feature space represents the co-visibility relations between local maps and extracted features for each view. We perform the correlation by employing a soft attention mechanism with raw features as guidance. Considering the fact that the capacity of recurrent units is finite, an attention module is operated in both spatial and temporal domains, as shown in Fig. 3.

More specifically, for each observation I_t, we take its raw feature map X_t as the guidance to select the most related knowledge by re-weighting all hidden states in the spatio-temporal domain. Since both X_t and H_i are 3D tensors, following GFS-VO [36], we calculate the cosine similarity between the vectorized feature map of X_t and H_i in each channel. Unlike GFS-VO that focuses on spatial correlation between two tensors, we extend the correlation to the temporal domain based on the intuition that each hidden state contributes discriminatively to different views. The whole process is described as:

$$X_t^{'} = \sum_{i=1}^{N} (\mathcal{A}^T(X_t, H_i) \sum_{j=1}^{C} \mathcal{A}^S(X_t^j, H_i^j) H_i^j) \ . \quad (2)$$

Here, $\mathcal{A}^T$ and $\mathcal{A}^S$ denote the temporal and spatial correlation respectively. N and C are the number of hidden states and channels each hidden state has. H_i^j denotes the jth channel of the ith hidden state. $X_t^{'} \in \mathbb{R}^{H \times W \times C}$ is the obtained input for I_t by fusing co-visible contents from the whole sequence explicitly in the feature space.

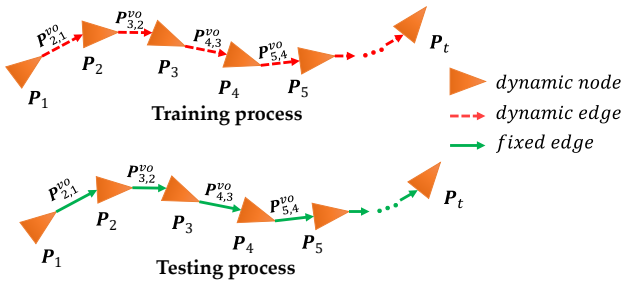


Figure 4: The pose graph consists of global poses indicating nodes and VO results representing edges. During training (top), both the nodes and edges are optimized and updated dynamically. While in testing (bottom), we perform a standard pose graph optimization [5] to optimize only the global poses with relative poses fixed as marginalization.

The soft attention we adopt discovers the filtered co-visible contents and retains observations beyond a single view but appear in local maps. More importantly, the obtained content for each frame is view related and the field of view is enlarged implicitly by images in the whole sequence in feature space.

Global Pose Estimation Given the augmented feature $X_t^{'}$ at time t, we can estimate the corresponding global pose with the *Estimator*. $X_t^{'}$ is fused by two convolutional layers with kernel size of 3. Then the feature is fed into a GAP layer and three FC layers (with output channels of 2048, 3 and 3), mapping features to global poses (see Fig. 2). As FC layers preserve much of the global map information [4], the process can be viewed as a registration between the *local map* from a specific view and the *global map* of the scene.

3.3. Motion-based Refinement

The VO estimation provides relative poses with promising accuracy but drifts over time. The predicted global poses undergo uncertainties while are drift-free. Therefore, the two sub-tasks can be learned cooperatively, resulting in a win-win situation. In our work, we employ a pose graph to enforce the consistency of predictions. In the graph, all global poses are defined as nodes which are connected by edges, representing the relative poses, as shown in Fig. 4.

Joint Optimization in Training During the training process, since the global and relative poses are not optimal, both nodes and edges are optimized and updated dynamically. We consider a local window with all frames in the videos, and design the joint loss with the global and local poses coupled as:

$$\mathcal{L}_{joint} = \sum_{i=1}^{N-1} \mathcal{D}(P_{i+1}, P_{i+1,i}^{vo} P_i) \ . \quad (3)$$

The $\mathcal{D}$ computes the distance between two poses (to be discussed further). $\mathcal{L}_{joint}$ indicates the consistency between predicted global and relative poses. As the joint loss mainly enforces the consistency between predicted poses, it is combined with additional losses to optimize the model in a supervised manner.

Pose Graph Optimization in Testing When in testing, the VO sub-network recovers locally accurate relative poses between consecutive frames. To further reduce the uncertainty of global poses, we perform a standard pose graph optimization (PGO) [5] by optimizing only the nodes with edges fixed. We minimize the error as:

$$\min_{P'_{1:N}} \sum_{i=1}^{N} \mathcal{D}(P_i, P'_i) + \alpha \sum_{i=1}^{N-1} \mathcal{D}(P'_{i+1}, P^{vo}_{i+1,i} P'_i) \,. \quad (4)$$

P_i and $P^{vo}_{i+1,i}$ are the predicted global and local poses. P'_i denotes global poses to be optimized. As no loop closures are provided as in classic SLAM systems [21], the predicted global poses are optimized with local poses as constraints to force the consistency. The error against original predictions is added, restricting the final results from deviating too much from the original predictions. α is used to balance the two errors.

Though MapNet [4] also employs a pose graph optimization during inference, it requires extra computation with known relative poses of testing images from the ground truth or other VO/SLAM systems [8]. In contrast, our approach obtains these results from the model, and thus gets rid of the dependence on extra data.

3.4. Loss Function

As our network learns camera relocalization and VO estimation jointly in a unified model, the losses are defined on both the global and relative poses as:

$$\mathcal{L}_g = \sum_{i=1}^{N} \mathcal{D}(\hat{P}_i, P_i) \,, \quad (5)$$

$$\mathcal{L}_{vo} = \sum_{i=2}^{N} \mathcal{D}(\hat{P}^{vo}_{i+1,i}, P^{vo}_{i+1,i}) \,. \quad (6)$$

$\mathcal{L}_g$ and $\mathcal{L}_{vo}$ represent the global and relative losses, respectively. $\hat{P}_i$ and $\hat{P}^{vo}_i$ are ground truth global and relative poses. For function $\mathcal{D}$, we adopt the definition in [15]:

$$\mathcal{D}(\hat{P}, P) = ||\hat{\mathbf{t}} - \mathbf{t}||_1 \exp^{-\beta} + \beta + ||\hat{\mathbf{w}} - \mathbf{w}||_1 \exp^{-\gamma} + \gamma \,, \quad (7)$$

where $\mathbf{t}$ and $\mathbf{w}$ are both 3-DoF parameters representing the predicted translation and rotation (in formulation of log quaternion), while $\hat{\mathbf{t}}$ and $\hat{\mathbf{w}}$ denote the ground truths. β and γ are the weights to balance the translational and rotational errors. During training, β and γ are not fixed, we optimize them as parameters with an initialization β^0 and γ^0.

The total loss consists of global pose loss $\mathcal{L}_g$, VO loss $\mathcal{L}_{vo}$ and motion constraint loss $\mathcal{L}_{joint}$ as:

$$\mathcal{L}_{total} = \mathcal{L}_g + \mathcal{L}_{vo} + \mathcal{L}_{joint} \,. \quad (8)$$

Although different losses have different scales, the parameters β and γ of each loss can balance the scale.

4. Experiments

In this section, we first introduce the implementation details, datasets used for evaluation and baseline methods for comparison. Then we compare our model with previous approaches in Sec. 4.1 and 4.2. The ablation study and additional results can be found in the supplementary material.

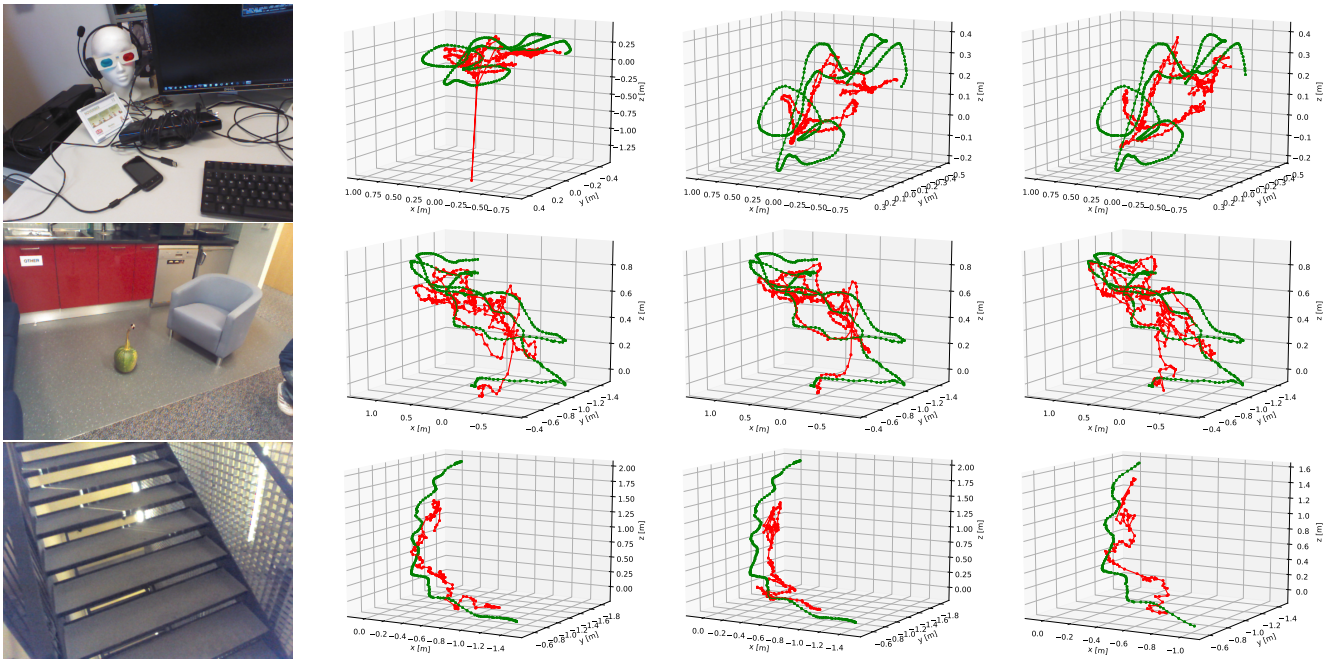
Implementation Details Our network takes monocular RGB image sequences as input. We use 7 consecutive images to construct a sequence. Similar to [4], shorter side length of all input images are rescaled to 256. We calculate the pixel-wise mean for each of the scenes in the datasets and subtract them with the input images. All βs and γs are set to -3 and 0, respectively. The ResNet34 is pretrained on the ImageNet, while other parts of the network are initialized using MSRA method [10]. We adopt the PyTorch [22] to implement the model on an NVIDIA 1080Ti GPU. Adam [17] with weight decay of 5×10^{-4} is used to optimize the network with batch size of 16 for 200 epochs in total. The initial learning rate is set to 10^{-4} and is kept during the whole training process.

Dataset We conduct extensive experiments on the popular 7-Scenes [29] and Oxford RobotCar [19] datasets. The 7-Scenes was collected using a Kinect in seven different indoor environments with spatial extent less than 4 meters. Among the 7 categories, the *Pumpkin* was recorded in the scene with a pumpkin on the floor, and contains many texture-less areas. The *Stairs* was collected over the stairs with lots of highly repetitive textures. Both the *Pumpkin* and *Stairs* are challenging for relocalization algorithms (see Fig. 5). Each category contains multiple sequences, some of which are used for training and the others for testing. The ground truth was obtained with KinectFusion. Both RGB and depth image sequences are provided, while only the RGB images are utilized in this paper.

The Oxford RobotCar dataset was captured through central Oxford over several periods in a year. Consequently, it contains observations under various conditions of weather, traffic, pedestrians, construction and roadworks, which makes it a big challenge for relocalization algorithms. Images captured by the center camera are used as input and the interpolations of INS measurements are used as the ground truth. We follow the train/split in MapNet [4] by using two groups of subsets denoted as LOOP and FULL scenes.

Method	Sequence							
	Chess	Fire	Heads	Office	Pumpkin	Kitchen	Stairs	Avg
PoseNet15 [16]	0.32 m, 8.12°	0.47m, 14.4°	0.29m, 12.0°	0.48m, 7.68°	0.47m, 8.42°	0.59m, 8.64°	0.47m, 13.8°	0.44m, 10.4°
PoseNet16 [14]	0.37 m, 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°
PoseNet17 [15]	0.13m, 4.48°	0.27m, 11.30°	0.17m, 13.00°	0.19m, 5.55°	0.26m, 4.75°	0.23m, 5.35°	0.35m, 12.40°	0.23m, 8.12°
Hourglass [20]	0.15m, 6.17°	0.27m, **10.84°**	0.19m, **11.63°**	0.21m, 8.48°	0.25m, 7.01°	0.27m, 10.15°	0.29m, 12.46°	0.23m, 9.53°
LSTM-Pose [34]	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°
Ours	**0.09**m, **3.28°**	**0.26**m, 10.92°	**0.17**m, 12.70°	**0.18**m, **5.45°**	**0.20**m, **3.66°**	**0.23**m, **4.92°**	**0.23**m, **11.3°**	**0.19**m, **7.47°**

Table 1: Median translation and rotation errors of previous methods using single images as input and our model on the 7-Scenes dataset [29]. The best results are highlighted.



Raw images PoseNet [14–16] MapNet [4] Ours

Figure 5: Results on the 7-Scenes dataset [29]. The green line indicates the ground truth and the red line represents the predicted trajectories of various mehtods. From top to bottom, three testing sequences are Heads-seq-01, Pumpkin-seq-07, Stairs-seq-01. The *Pumpkin* contains many texture-less regions, and *Stairs* has lots of highly repetitive textures.

Baseline Methods When evaluating on the 7-Scenes benchmark, the baseline algorithms include PoseNet15 [16], PoseNet16 [14], PoseNet17 [15], Hourglass [20], LSTM-PoseNet [34], VidLoc [7], and MapNet [4]. Both PoseNet15 [16] and MapNet [4] are used for comparison on the Oxford RobotCar. Although VidLoc [7] reported results on the LOOP scene, the training/testing sequences are not provided.

4.1. Experiments on the 7-Scenes Dataset

Comparison with Image-based Methods Table 1 shows the results of methods taking single images as input and our model. Obviously, our approach outperforms these methods by a large margin. As these algorithms rely on single images to recover global poses, their results inevitably suffer from high uncertainties.

Comparison with Sequence-based Methods Table 2 demonstrates the comparison against sequence-based relocalization methods including VidLoc [7], MapNet [4] and DSO [8]. Our approach achieves slightly better performance on regular scenes including *Fire*, *Heads*, *Kitchen*. While on scenarios with many texture-less regions (*Pumpkin*) and highly repetitive textures (*Stairs*), our model obtains outstanding results. As VidLoc [7] only considers the fusion of observations, thus achieves lower accuracy in translation. MapNet [4] introduces the pose constraints during the training process and achieves promising performance on regular scenes, while our model gives results with higher accuracy. More importantly, our methods shows great potential in dealing with difficult scenarios.

Scene	Method			
	DSO - [8]	VidLoc [7]	MapNet [4]	**Ours**
Chess	0.17m, 8.13°	0.18m, NA	**0.08**m, **3.25**°	0.09m, 3.28°
Fire	0.19m, 65.0°	0.26m, NA	0.27m, 11.69°	**0.26**m, **10.92**°
Heads	0.61m, 68.2°	**0.14**m, NA	0.18m, 13.25°	0.17m, **12.70**°
Office	1.51m, 16.8°	0.26m, NA	**0.17**m, **5.15**°	0.18m, 5.45°
Pumpkin	0.61m, 15.8°	0.36m, NA	0.22m, 4.02°	**0.20**m, **3.69**°
Kitchen	0.23m, 10.9°	0.31m, NA	0.23m, 4.93°	**0.23**m, **4.92**°
Stairs	0.26m, 21.3°	0.26m, NA	0.30m, 12.08°	**0.23**m, **11.3**°
Avg	0.26m, 29.4°	0.25m, NA	0.21m, 7.77°	**0.19**m, **7.47**°

Table 2: Median translation and rotation errors of DSO [8], VidLoc [7], MapNet [4] and our approach on the 7-Scenes dataset [29]. The best results are highlighted.

Fig. 5 illustrates the qualitative comparison between previous methods and our approach. As PoseNet and its variations [14–16] localize the camera from single images individually, they produce many outliers in both regular and challenging scenes. The number of outliers are reduced by introducing motion constraints over outputs as MapNet [4]. However, this strategy brings finite improvements on scenes with texture-less regions and repetitive textures, resulting in zigzag trajectories. By considering the spatio-temporal consistency of sequential images, our model gains much smoother trajectories in the *Pumpkin* and *Stairs* categories.

4.2. Experiments on the Oxford RobotCar Dataset

We evaluate our model on the LOOP and FULL routes of the Oxford RobotCar dataset [19]. Both of the two routes are very long with trajectories of approximate 1120m and 9562m. The training and testing sequences are captured under various weather conditions, which makes them laborious for localization methods. Other uncertainties including dynamic objects, similar appearances, and over exposure further increase the difficulty, as shown in Fig. 6.

Quantitative Comparison Table 3 shows the quantitative comparison of our approach against PoseNet and MapNet. Since the training and testing sequences are captured under varying conditions at different times, PoseNet can hardly handle such changes and outputs poor poses. MapNet produces promising results on shorter routes (LOOP1 and LOOP2), but the error increases sharply with the scene growing larger (from 1120m to 9562m). The larger areas may contain more locally similar appearances, degrading the ability the relocalization systems. In contrast, taking both the content and motion into consideration, our model deals with these challenges more effectively.

Qualitative Comparison Fig. 7 represents the trajectories recovered by Stereo VO (results from [19]), PoseNet, MapNet and our model. As shown clearly, Stereo VO produces very smooth trajectories as the relative poses between consecutive frames can be calculated locally accurately. Yet, Stereo VO drifts severely over time in such long

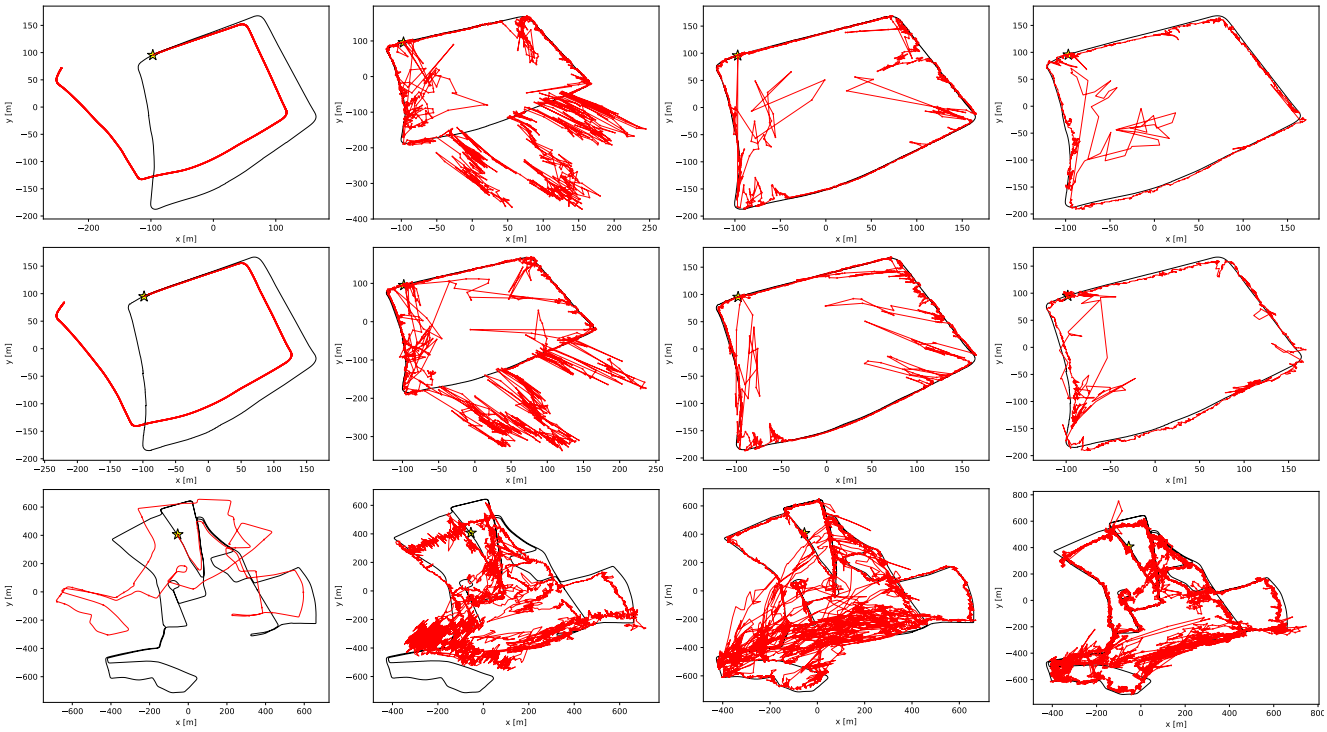
Figure 6: Samples of challenging conditions on the RobotCar dataset [19]. From left to right: three pairs of images demonstrate the observations with over-exposure, changing weather and dynamic objects, similar appearances at different locations.

Scene	Method		
	PoseNet [14–16]	MapNet [4]	**Ours**
LOOP1	28.81m, 19.62°	**8.76**m, 3.46°	9.07m, **3.31**°
LOOP2	25.29m, 17.45°	9.84m, 3.96°	**9.19**m, **3.53**°
FULL1	125.6m, 27.1°	41.4m, 12.5°	**31.65**m, **4.51**°
FULL2	131.06m, 26.05°	59.30m, 14.81°	**53.45**m, **8.60**°
Avg	77.85m, 22.56°	29.83m, 8.68°	**25.84**m, **4.99**°

Table 3: Mean translation and rotation errors of PoseNet [14–16], MapNet [4] nad our method on Oxford RobotCar dataset [19]. Results of PoseNet and MapNet are from [4]. The best results are highlighted.

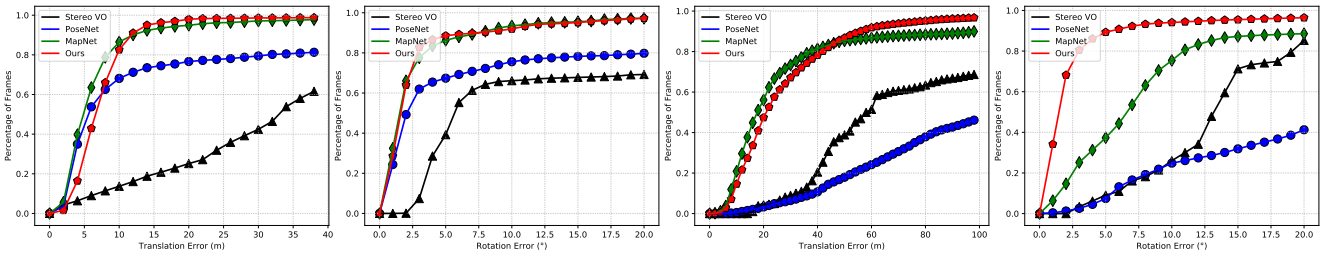
routes. PoseNet predicts a lot of outliers which are far from the ground truth locations. Utilizing the motion consistency, MapNet [4] improves the accuracy of predicted poses and reduces the number of outliers by a large margin. However, there are still a lot of outliers across the whole areas due to the local similarities of observations in large scenarios. Compared with PoseNet and MapNet, our model eliminates outliers more effectively and performs more stably in these areas where the previous two methods give noisy results.

Cumulative Distribution Errors We additionally calculate the cumulative distribution of the translation and rotation errors on the LOOP and FULL scenes (shown in Fig. 8). We can find that Stereo VO and PoseNet produce poor results in both translation and rotation. MapNet and our method produce very close results in regular regions. While for scenarios causing larger uncertainties, the performance of our model is markedly more stable. Deep learning techniques are not good at estimating camera orientations [25], let alone from single images. Therefore, another important advantage of our model is its ability to give accurate orientation estimation by considering the local information of sequential images.



Stereo VO [19] PoseNet [14–16] MapNet [4] Ours

Figure 7: Results of Stereo VO, PoseNet, MapNet and our model on the LOOP1 (top), LOOP2 (middle) and FULL1 scenes (bottom) of the Oxford RobotCar dataset [19] The red and black lines indicate predicted and ground truth poses respectively.



(a) Translation error on LOOP. (b) Rotation error on LOOP. (c) Translation error on FULL. (d) Rotation error on FULL.

Figure 8: Cumulative distributions of the mean translation (m) and rotation errors (○) of all the methods evaluated on Oxford RobotCar LOOP and FULL scenes. X-axis is the errors and y-axis is the percentage of frames with error less than the value.

5. Conclusion

In this paper, we aim to overcome the ambiguities of single images in relocalization with assistance of the local information in image sequences. To fully use the spatio-temporal consistency, we incorporate a VO component. Specifically, instead of raw features, our model estimates global poses from features which are augmented by local maps preserved in recurrent units based on the co-visibility. Besides, as relative poses can be calculated with promising accuracy in VO component due to the continuity of camera motions, the predicted global poses are further optimized with these relative poses constraints within a pose graph during both the training and testing processes. Experiments on both the indoor 7-Scenes and outdoor Oxford RobotCar datasets demonstrate that our approach outperforms previous methods, especially in challenging scenarios.

Acknowledgement

The work is supported by the National Key Research and Development Program of China (2017YFB1002601) and National Natural Science Foundation of China (61632003, 61771026).

References

[1] Vassileios Balntas, Shuda Li, and Victor Prisacariu. RelocNet: Continuous Metric Learning Relocalisation Using Neural Nets. In *ECCV*, 2018.

[2] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. DSAC-differentiable RANSAC for Camera Localization. In *CVPR*, 2017.

[3] Eric Brachmann and Carsten Rother. Learning Less Is More - 6D Camera Localization via 3D Surface Regression. In *CVPR*, 2018.

[4] Samarth Brahmbhatt, Jinwei Gu, Kihwan Kim, James Hays, and Jan Kautz. MapNet: Geometry-aware Learning of Maps for Camera Localization. In *CVPR*, 2018.

[5] Giuseppe Calafiore, Luca Carlone, and Frank Dellaert. Pose Graph Optimization in the Complex Domain: Lagrangian Duality, Conditions for Zero Duality Gap, and Optimal Solutions. *T-RO*, 2016.

[6] Federico Camposeco, Torsten Sattler, Andrea Cohen, Andreas Geiger, and Marc Pollefeys. Toroidal Constraints for Two-point Localization under High Outlier Ratios. In *CVPR*, 2017.

[7] Ronald Clark, Sen Wang, Andrew Markham, Niki Trigoni, and Hongkai Wen. Vidloc: A Deep Spatio-temporal Model for 6-DOF Video-clip Relocalization. In *CVPR*, 2017.

[8] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct Sparse Odometry. *TPAMI*, 2018.

[9] Jakob Engel, Thomas Schöps, and Daniel Cremers. LSD-SLAM: Large-scale Direct Monocular SLAM. In *ECCV*, 2014.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving Deep into Rectifiers: Surpassing Human-level Performance on Imagenet Classification. In *ICCV*, 2015.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016.

[12] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-term Memory. *Neural Computation*, 1997.

[13] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating Local Descriptors into a Compact Image Representation. In *CVPR*, 2010.

[14] Alex Kendall and Roberto Cipolla. Modelling Uncertainty in Deep Learning for Camera Relocalization. In *ICRA*, 2016.

[15] Alex Kendall and Roberto Cipolla. Geometric Loss Functions for Camera Pose Regression with Deep Learning. In *CVPR*, 2017.

[16] Alex Kendall, Matthew Grimes, and Roberto Cipolla. PoseNet: A Convolutional Network for Real-time 6-DoF Camera Relocalization. In *ICCV*, 2015.

[17] Diederik P Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *ICLR*, 2015.

[18] Liu Liu, Hongdong Li, and Yuchao Dai. Efficient Global 2D-3D Matching for Camera Localization in a Large-scale 3D Map. In *ICCV*, 2017.

[19] Will Maddern, Geoff Pascoe, Chris Linegar, and Paul Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *IJRR*, 2017.

[20] Iaroslav Melekhov, Juha Ylioinas, Juho Kannala, and Esa Rahtu. Image-based Localization Using Hourglass Networks. In *ICCV Workshops*, 2017.

[21] Raul Mur-Artal and Juan D Tardós. ORB-SLAM2: An Open-source SLAM System for Monocular, Stereo, and RGB-D Cameras. *T-RO*, 2017.

[22] Adam Paszke, Sam Gross, Soumith Chintala, and Gregory Chanan. Pytorch. `https://github.com/pytorch/pytorch`, 2017.

[23] Torsten Sattler, Michal Havlena, Konrad Schindler, and Marc Pollefeys. Large-scale Location Recognition and the Geometric Burstiness Problem. In *CVPR*, 2016.

[24] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Efficient & Effective Prioritized Matching for Large-scale Image-based Localization. *TPAMI*, 2017.

[25] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, Fredrik Kahl, and Tomas Pajdla. Benchmarking 6-DoF Outdoor Visual Localization in Changing Conditions. In *CVPR*, 2018.

[26] Torsten Sattler, Qunjie Zhou, Marc Pollefeys, and Laura Leal-Taixe. Understanding the Limitations of CNN-based Absolute Camera Pose Regression. In *CVPR*, 2019.

[27] Johannes L Schonberger and Jan Michael Frahm. Structure-from-motion Revisited. In *CVPR*, 2016.

[28] Xingjian Shi, Zhourong Chen, Hao Wang, Dit Yeung, Wai Wong, and Wang Woo. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. In *NIPS*, 2015.

[29] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene Coordinate Regression Forests for Camera Relocalization in RGB-D Images. In *CVPR*, 2013.

[30] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. End-to-end Memory Networks. In *NIPS*, 2015.

[31] Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Akihiko Torii. InLoc: Indoor Visual Localization with Dense Matching and View Synthesis. In *CVPR*, 2018.

[32] Carl Toft, Erik Stenborg, Lars Hammarstrand, Lucas Brynte, Marc Pollefeys, Torsten Sattler, and Fredrik Kahl. Semantic Match Consistency for Long-term Visual Localization. In *ECCV*, 2018.

[33] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. DeMoN: Depth and Motion Network for Learning Monocular Stereo. In *CVPR*, 2017.

[34] Florian Walch, Caner Hazirbas, Laura Leal-Taixe, Torsten Sattler, Sebastian Hilsenbeck, and Daniel Cremers. Image-based Localization Using LSTMs for Structured Feature Correlation. In *ICCV*, 2017.

[35] Sen Wang, Ronald Clark, Hongkai Wen, and Niki Trigoni. DeepVO: Towards End-to-end Visual Odometry with Deep Recurrent Convolutional Neural Networks. In *ICRA*, 2017.

[36] Fei Xue, Qiuyuan Wang, Xin Wang, Wei Dong, Junqiu Wang, and Hongbin Zha. Guided Feature Selection for Deep Visual Odometry. In *ACCV*, 2018.

[37] Fei Xue, Xin Wang, Shunkai Li, Qiuyuan Wang, Junqiu Wang, and Hongbin Zha. Beyond Tracking: Selecting Memory and Refining Poses for Deep Visual Odometry. In *CVPR*, 2019.

[38] Zhichao Yin and Jianping Shi. GeoNet: Unsupervised Learning of Dense Depth, Optical Flow and Camera Pose. In *CVPR*, 2018.

[39] Huizhong Zhou, Benjamin Ummenhofer, and Thomas Brox. DeepTAM: Deep Tracking and Mapping. In *ECCV*, 2018.

[40] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised Learning of Depth and Ego-motion from Video. In *CVPR*, 2017.