KFNet: Learning Temporal Camera Relocalization using Kalman Filtering

Lei Zhou[1] Zixin Luo[1] Tianwei Shen[1] Jiahui Zhang[2]
Mingmin Zhen[1] Yao Yao[1] Tian Fang[3] Long Quan[1]
[1]Hong Kong University of Science and Technology [2]Tsinghua University [3]Everest Innovation Technology
[1]{lzhouai,zluoag,tshenaa,mzhen,yyaoag,quan}@cse.ust.hk
[2]jiahui-z15@mails.tsinghua.edu.cn [3]fangtian@altizure.com

Abstract

Temporal camera relocalization estimates the pose with respect to each video frame in sequence, as opposed to one-shot relocalization which focuses on a still image. Even though the time dependency has been taken into account, current temporal relocalization methods still generally underperform the state-of-the-art one-shot approaches in terms of accuracy. In this work, we improve the temporal relocalization method by using a network architecture that incorporates Kalman filtering (KFNet) for online camera relocalization. In particular, KFNet extends the scene coordinate regression problem to the time domain in order to recursively establish 2D and 3D correspondences for the pose determination. The network architecture design and the loss formulation are based on Kalman filtering in the context of Bayesian learning. Extensive experiments on multiple relocalization benchmarks demonstrate the high accuracy of KFNet at the top of both one-shot and temporal relocalization approaches. Our codes are released at https://github.com/zlthinker/KFNet.

1. Introduction

Camera relocalization serves as the subroutine of applications including SLAM [15], augmented reality [9] and autonomous navigation [45]. It estimates the 6-DoF pose of a query RGB image in a known scene coordinate system. Current relocalization approaches mostly focus on one-shot relocalization for a still image. They can be mainly categorized into three classes [13, 50]: (1) the relative pose regression (RPR) methods which determine the relative pose w.r.t. the database images [3, 29], (2) the absolute pose regression (APR) methods regressing the absolute pose through PoseNet [25] and its variants [23, 24, 60] and (3) the structure-based methods that establish 2D-3D correspondences with Active Search [48, 49] or Scene Coordinate Regression (SCoRe) [52] and then solve the pose by PnP algorithms [18, 42]. Particularly, SCoRe is widely adopted recently to learn per-pixel scene coordinates from dense training data for a scene, due to its ability to form dense and accurate 2D-3D matches even in texture-less scenes [5, 6]. As extensively evaluated in [5, 6, 50], the structure-based methods generally show better pose accuracy than the RPR and APR methods, because they explicitly exploit the rules of the projective geometry and the scene structures [50].

Apart from one-shot relocalization, temporal relocalization with respect to video frames is also worthy of investigation. However, almost all the temporal relocalization methods are based on PoseNet [25], which, in general, even underperform the structure-based one-shot methods in accuracy. This is mainly because their accuracies are fundamentally limited by the retrieval nature of PoseNet. As analyzed in [50], PoseNet based methods are essentially analogous to approximate pose estimation via image retrieval, and cannot go beyond the retrieval baseline in accuracy.

In this work, we are motivated by the high accuracy of structure-based relocalization methods and resort to SCoRe to estimate per-pixel scene coordinates for pose computation. Besides, we propose to extend SCoRe to the time domain in a recursive manner to enhance the temporal consistency of 2D-3D matching, thus allowing for more accurate online pose estimations for sequential images. Specifically, a recurrent network named *KFNet* is proposed in the context of Bayesian learning [37] by embedding SCoRe into the Kalman filter within a deep learning framework. It is composed of three subsystems below, as illustrated in Fig. 1.

- *The measurement system* features a network termed *SCoordNet* to derive the maximum likelihood (ML) predictions of the scene coordinates for a single image.
- *The process system* uses *OFlowNet* that models the optical flow based transition process for image pixels across time steps and yields the prior predictions of scene coordinates. Additionally, the measurement and process systems provide uncertainty predictions [40, 23] to model the noise dynamics over time.
- *The filtering system* fuses both predictions and leads to the maximum a posteriori (MAP) estimations of the final scene coordinates.

Furthermore, we propose probabilistic losses for the three subsystems based on the Bayesian formulation of KFNet, to enable the training of either the subsystems or the full framework. We summarize the contributions as follows.

- We are the first to extend the scene coordinate regression problem [52] to the time domain in a learnable way for temporally-consistent 2D-3D matching.
- We integrate the traditional Kalman filter [22] into a recurrent CNN network (KFNet) that resolves pixel-level state inference over time-series images.
- KFNet bridges the existing performance gap between temporal and one-shot relocalization approaches, and achieves top accuracy on multiple relocalization benchmarks [52, 57, 25, 43].
- Lastly, for better practicality, we propose a statistical assessment tool to enable KFNet to self-inspect the potential outlier predictions on the fly.

2. Related Works

Camera relocalization. We categorize camera relocalization algorithms into three classes: the relative pose regression (RPR) methods, the absolute pose regression (APR) methods and the structure-based methods.

The RPR methods use a coarse-to-fine strategy which first finds similar images in the database through image retrieval [55, 2] and then computes the relative poses w.r.t. the retrieved images [3, 29, 46]. They have good generalization to unseen scenes, but the retrieval process needs to match the query image against all the database images, which can be costly for time-critical applications.

The APR methods include PoseNet [25] and its variants [23, 24, 60] which learn to regress the absolute camera poses from the input images through a CNN. They are simple and efficient, but generally fall behind the structure-based methods in terms of accuracy, as validated by [5, 6, 50]. Theoretically, [50] explains that PoseNet-based methods are more closely related to image retrieval than to accurate pose estimation via 3D geometry.

The structure-based methods explicitly establish the correspondences between 2D image pixels and 3D scene points and then solve camera poses by PnP algorithms [18, 42, 30]. Traditionally, correspondences are searched by matching the patch features against Structure from Motion (SfM) tracks via Active Search [48, 49] and its variants [32, 10, 33, 47], which can be inefficient and fragile in textureless scenarios. Recently, the correspondence problem is resolved by predicting the scene coordinates for pixels by training random forests [52, 58, 36] or CNNs [5, 6, 31, 7] with ground truth scene coordinates, which is referred to as Scene Coordinate Regression (SCoRe).

Besides one-shot relocalization, some works have extended PoseNet to the time domain to address temporal relocalization. VidLoc [11] performs offline and batch relocalization for fixed-length video-clips by BLSTM [51]. Coskun *et al*. refine the pose dynamics by embedding LSTM units in the Kalman filters [12]. VLocNet [56] and VLocNet++ [43] propose to learn pose regression and the visual odometry jointly. LSG [63] combines LSTM with visual odometry to further exploit the spatial-temporal consistency. Since all the methods are extensions of PoseNet, their accuracies are fundamentally limited by the retrieval nature of PoseNet, following the analysis of [50].

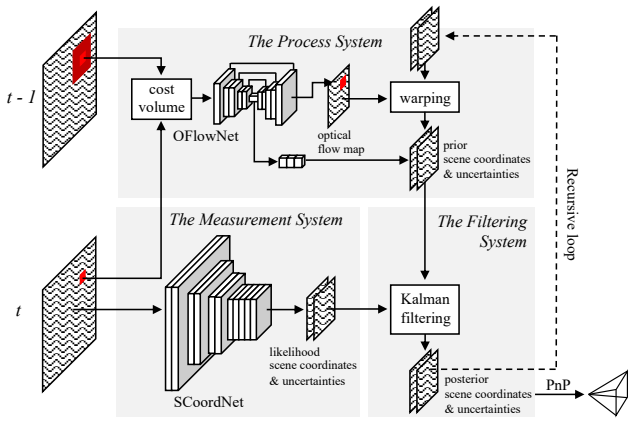


Figure 1: The architecture of the proposed KFNet, which is decomposed into the process, measurement and filtering systems.

Temporal processing. When processing time-series image data, ConvLSTM [61] is a standard way of modeling the spatial correlations of local contexts through time [59, 35, 28]. However, some works have pointed out that the implicit convolutional modeling is less suited to discovering the pixel associations between neighboring frames, especially when pixel-level accuracy is desired [21, 39]. Therefore, in later works, the optical flow is highlighted as a more explicit way of delineating the pixel correspondences across sequential steps [41]. For example, [41, 20, 28, 53, 39] commonly predict the optical flow fields to guide the feature map warping across time steps. Then, the warped features are fused by weighting [65, 66] or pooling [38, 41] to aggregate the temporal knowledge. In this work, we follow the practice of flow-guided warping, but the distinction from previous works is that we propose to fuse the predictions by leveraging Kalman filter principles [37].

3. Bayesian Formulation

This section presents the Bayesian formulation of recursive scene coordinate regression in the time domain for temporal camera relocalization. Based on the formulation, the proposed KFNet is built and the probabilistic losses are defined in Sec. 4 $\sim$ 6. Notations used below have been summarized in Table 1 for quick reference.

Given a stream of RGB images up to time t, *i.e*., $\mathcal{I}_t = \{\mathbf{I}_1, ..., \mathbf{I}_{t-1}, \mathbf{I}_t\}$, our aim is to predict the latent state for

Module	inputs	outputs	
The process system	$\hat{\boldsymbol{\theta}}_{t-1}$ $\boldsymbol{\Sigma}_{t-1}$ $\mathbf{I}_{t-1}$ $\mathbf{I}_t$	$\mathbf{G}_t$ $\mathbf{W}_t$ $\hat{\boldsymbol{\theta}}_t^- = \mathbf{G}_t\hat{\boldsymbol{\theta}}_{t-1}$ $\mathbf{R}_t = \mathbf{G}_t\boldsymbol{\Sigma}_{t-1}\mathbf{G}_t^T + \mathbf{W}_t$	*- transition matrix* *- process noise covariance* *- prior state mean* *- prior state covariance*
The measurement system	$\mathbf{I}_t$	$\mathbf{z}_t$ $\mathbf{V}_t$	*- state observations* *- measurement noise covariance*
The filtering system	$\hat{\boldsymbol{\theta}}_t^-$ $\mathbf{z}_t$ $\mathbf{R}_t$ $\mathbf{V}_t$	$\mathbf{e}_t = \mathbf{z}_t - \hat{\boldsymbol{\theta}}_t^-$ $\mathbf{K}_t = \frac{\mathbf{R}_t}{\mathbf{V}_t+\mathbf{R}_t}$ $\hat{\boldsymbol{\theta}}_t = \hat{\boldsymbol{\theta}}_t^- + \mathbf{K}_t\mathbf{e}_t$ $\boldsymbol{\Sigma}_t = \mathbf{R}_t(\mathbf{I}-\mathbf{K}_t)$	*- innovation* *- Kalman gain* *- posterior state mean* *- posterior state covariance*

Table 1: The summary of variables and notations used in the Bayesian formulation of KFNet.

each frame, *i.e.*, the scene coordinate map, which is then used for pose computation. We denote the map as $\boldsymbol{\theta}_t \in \mathbb{R}^{N\times 3}$, where N is the pixel number. By imposing the Gaussian noise assumption on the states, the state $\boldsymbol{\theta}_t$ conditioned on $\mathcal{I}_t$ follows an unknown Gaussian distribution:

$$\boldsymbol{\theta}_t^+ \overset{\text{def}}{=} (\boldsymbol{\theta}_t|\mathcal{I}_t) \sim \mathcal{N}(\hat{\boldsymbol{\theta}}_t, \boldsymbol{\Sigma}_t), \tag{1}$$

where $\hat{\boldsymbol{\theta}}_t$ and $\boldsymbol{\Sigma}_t$ are the expectation and covariance to be determined. Under the routine of Bayesian theorem, the posterior probability of $\boldsymbol{\theta}_t$ can be factorized as

$$P(\boldsymbol{\theta}_t|\mathcal{I}_t) \propto P(\boldsymbol{\theta}_t|\mathcal{I}_{t-1})P(\mathbf{I}_t|\boldsymbol{\theta}_t, \mathcal{I}_{t-1}), \tag{2}$$

where $\mathcal{I}_t = \mathcal{I}_{t-1} \cup \{\mathbf{I}_t\}$.

The first factor $P(\boldsymbol{\theta}_t|\mathcal{I}_{t-1})$ of the right hand side (RHS) of Eq. 2 indicates the prior belief about $\boldsymbol{\theta}_t$ obtained from time $t-1$ through a *process system*. Provided that no occlusions or dynamic objects occur, the consecutive coordinate maps can be approximately associated by a linear **process equation** describing their pixel correspondences, wherein

$$\boldsymbol{\theta}_t = \mathbf{G}_t\boldsymbol{\theta}_{t-1} + \mathbf{w}_t, \tag{3}$$

with $\mathbf{G}_t \in \mathbb{R}^{N\times N}$ being the sparse state transition matrix given by the optical flow fields from time $t-1$ to t, and $\mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{W}_t)$, $\mathbf{W}_t \in \mathbb{S}_{++}^N$[1] being the *process noise*. Given $\mathcal{I}_{t-1}$, we already have the probability statement that $(\boldsymbol{\theta}_{t-1}|\mathcal{I}_{t-1}) \sim \mathcal{N}(\hat{\boldsymbol{\theta}}_{t-1}, \boldsymbol{\Sigma}_{t-1})$. Then the prior estimation of $\boldsymbol{\theta}_t$ from time $t-1$ can be expressed as

$$\boldsymbol{\theta}_t^- \overset{\text{def}}{=} (\boldsymbol{\theta}_t|\mathcal{I}_{t-1}) \sim \mathcal{N}(\hat{\boldsymbol{\theta}}_t^-, \mathbf{R}_t), \tag{4}$$

where $\hat{\boldsymbol{\theta}}_t^- = \mathbf{G}_t\hat{\boldsymbol{\theta}}_{t-1}$, $\mathbf{R}_t = \mathbf{G}_t\boldsymbol{\Sigma}_{t-1}\mathbf{G}_t^T + \mathbf{W}_t$.

The second factor $P(\mathbf{I}_t|\boldsymbol{\theta}_t, \mathcal{I}_{t-1})$ of the RHS of Eq. 2 describes the likelihood of image observations at time t made through a *measurement system*. The system models how $\mathbf{I}_t$ is derived from the latent states $\boldsymbol{\theta}_t$, formally $\mathbf{I}_t = \mathbf{h}(\boldsymbol{\theta}_t)$. However, the high nonlinearity of $\mathbf{h}(\cdot)$ makes the following computation intractable. Alternatively, we map $\mathbf{I}_t$ to $\mathbf{z}_t \in \mathbb{R}^{N\times 3}$ via a nonlinear function inspired by [12], so that the system can be approximately expressed by a linear **measurement equation**:

$$\mathbf{z}_t = \boldsymbol{\theta}_t + \mathbf{v}_t, \tag{5}$$

where $\mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{V}_t)$, $\mathbf{V}_t \in \mathbb{S}_{++}^N$ denotes the *measurement noise*, and $\mathbf{z}_t$ can be interpreted as the noisy observed scene coordinates. In this way, the likelihood can be rewritten as $P(\mathbf{z}_t|\boldsymbol{\theta}_t, \mathcal{I}_{t-1})$ by substituting $\mathbf{z}_t$ for $\mathbf{I}_t$.

Let $\mathbf{e}_t$ denote the residual of predicting $\mathbf{z}_t$ from time $t-1$; thus

$$\mathbf{e}_t = \mathbf{z}_t - \hat{\boldsymbol{\theta}}_t^- = \mathbf{z}_t - \mathbf{G}_t\hat{\boldsymbol{\theta}}_{t-1}. \tag{6}$$

Since $\mathbf{G}_t$ and $\hat{\boldsymbol{\theta}}_{t-1}$ are all known, observing $\mathbf{z}_t$ is equivalent to observing $\mathbf{e}_t$. Hence, the likelihood $P(\mathbf{z}_t|\boldsymbol{\theta}_t, \mathcal{I}_{t-1})$ can be rewritten as $P(\mathbf{e}_t|\boldsymbol{\theta}_t, \mathcal{I}_{t-1})$. Substituting Eq. 5 into Eq. 6, we have $\mathbf{e}_t = \boldsymbol{\theta}_t - \hat{\boldsymbol{\theta}}_t^- + \mathbf{v}_t$, so that the likelihood can be described by

$$(\mathbf{e}_t|\boldsymbol{\theta}_t, \mathcal{I}_{t-1}) \sim \mathcal{N}(\boldsymbol{\theta}_t - \hat{\boldsymbol{\theta}}_t^-, \mathbf{V}_t). \tag{7}$$

Based on the theorems in multivariate statistics [1, 37], combining the two distributions 4 & 7 gives the bivariate normal distribution:

$$\left[\begin{pmatrix}\boldsymbol{\theta}_t\\ \mathbf{e}_t\end{pmatrix}\middle|\,\mathcal{I}_{t-1}\right] \sim \mathcal{N}\left[\begin{pmatrix}\hat{\boldsymbol{\theta}}_t^-\\ \mathbf{0}\end{pmatrix}, \begin{pmatrix}\mathbf{R}_t & \mathbf{R}_t\\ \mathbf{R}_t & \mathbf{R}_t + \mathbf{V}_t\end{pmatrix}\right]. \tag{8}$$

Making $\mathbf{e}_t$ the conditioning variable, *the filtering system* gives the posterior distribution that writes

$$\begin{aligned}\boldsymbol{\theta}_t^+ \overset{\text{def}}{=} (\boldsymbol{\theta}_t|\mathcal{I}_t) = (\boldsymbol{\theta}_t|\mathbf{e}_t, \mathcal{I}_{t-1}) &\sim \mathcal{N}(\hat{\boldsymbol{\theta}}_t, \boldsymbol{\Sigma}_t)\\ &\sim \mathcal{N}(\hat{\boldsymbol{\theta}}_t^- + \mathbf{K}_t\mathbf{e}_t, \mathbf{R}_t(\mathbf{I}-\mathbf{K}_t)),\end{aligned} \tag{9}$$

where $\mathbf{K}_t = \frac{\mathbf{R}_t}{\mathbf{V}_t+\mathbf{R}_t}$ is conceptually referred to as the *Kalman gain* and $\mathbf{e}_t$ as the *innovation*[2] [37, 19].

As shown in Fig. 1, the inference of the posterior scene coordinates $\hat{\boldsymbol{\theta}}_t$ and covariance $\boldsymbol{\Sigma}_t$ for image pixels proceeds recursively as the time t evolves, which are then used for online pose determination. Specifically, the pixels with variances greater than λ are first excluded as outliers. Then, a RANSAC+P3P [18] solver is applied to compute the initial camera pose from the 2D-3D correspondences, followed by a nonlinear optimization for pose refinement.

4. The Measurement System

The measurement system is basically a generative model explaining how the observations $\mathbf{z}_t$ are generated from the latent scene coordinates $\boldsymbol{\theta}_t$, as expressed in Eq. 5. Then, the remaining problem is to learn the underlying mapping from $\mathbf{I}_t$ to $\mathbf{z}_t$. This is similar to the SCoRe task [52, 5, 6], but differs in the constraint about $\mathbf{z}_t$ imposed by Eq. 5. Below, the

[1] $\mathbb{S}_{++}^N$ denotes the set of N-dimensional positive definite matrices.

[2] The derivation of Eqs. 8 & 9 is shown in the supplementary material.

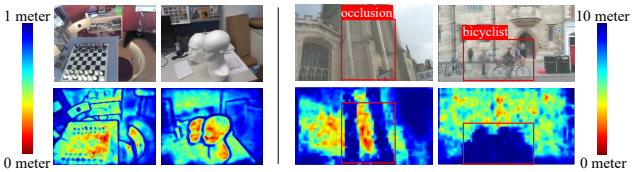


Figure 2: The visualization of uncertainties which model the measurement noise and the process noise. (a) SCoordNet predicts larger uncertainties from single images over the object boundaries where larger errors occur. (b) OFlowNet gives larger uncertainties from the consecutive images (overlaid) over the areas where occlusions or dynamic objects appear.

architecture of SCoordNet is first introduced, which outputs the scene coordinate predictions, along with the uncertainties, to model the measurement noise $\mathbf{v}_t$. Then, we define the probabilistic loss based on the likelihood $P(\mathbf{z}_t|\boldsymbol{\theta}_t, \mathcal{I}_{t-1})$ of the measurement system.

4.1. Architecture

SCoordNet shares the similar fully convolutional structure to [6], as shown in Fig. 1. However, it is far more lightweight, with parameters fewer than one eighth of [6]. It encompasses twelve 3×3 convolution layers, three of which use a stride of 2 to downsize the input by a factor of 8. ReLU follows each layer except the last one. To simplify computation and avoid the risk of over-parameterization, we postulate the isotropic covariance of the multivariate Gaussian measurement noise, *i.e.*, $\mathbf{V}_{(i)} = {v_{(i)}}^2\mathbb{I}_3$ for each pixel $\mathbf{p}_i$, where $\mathbb{I}_3$ denotes the 3×3 identity matrix. The output thus has a channel of 4, comprising 3-d scene coordinates and a 1-d uncertainty measurement.

4.2. Loss

According to Eq. 5, the latent scene coordinates $\boldsymbol{\theta}_{(i)}$ of pixel $\mathbf{p}_i$ should follow the distribution $\mathcal{N}(\mathbf{z}_{(i)}, v_{(i)}^2\mathbb{I}_3)$. Taking the negative logarithm of the probability density function (PDF) of $\boldsymbol{\theta}_{(i)}$, we define the loss based on the likelihood which gives rise to the maximum likelihood (ML) estimation for each pixel in the form [23]:

$$\mathcal{L}_{likelihood} = \sum_{i=1}^{N}\left(3\log v_{(i)} + \frac{\|\mathbf{z}_{(i)} - \mathbf{y}_{(i)}\|_2^2}{2{v_{(i)}}^2}\right), \quad (10)$$

with $\mathbf{y}_{(i)}$ being the groundtruth label for $\boldsymbol{\theta}_{(i)}$. For numerical stability, we use logarithmic variance for the uncertainty measurements in practice, *i.e.*, $s_{(i)} = \log {v_{(i)}}^2$.

Including uncertainty learning in the loss formulation allows one to quantify the prediction errors stemming not just from the intrinsic noise in the data but also from the defined model [14]. For example, at the boundary with depth discontinuity, a sub-pixel offset would cause an abrupt coordinate shift which is hard to model. SCoordNet would easily suffer from a significant magnitude of loss in such cases. It is sensible to automatically downplay such errors during training by weighting with the uncertainty measurements. Fig. 2(a) illustrates the uncertainty predictions in such cases.

5. The Process System

The process system models the transition process of pixel states from time $t-1$ to t, as described by the process equation of Eq. 3. Herein, first, we propose a cost volume based network, OFlowNet, to predict the optical flows and the process noise covariance jointly for each pixel. Once the optical flows are determined, Eq. 3 is equivalent to the flow-guided warping from time $t-1$ towards t, as commonly used in [41, 20, 28, 53, 39]. Second, after the warping, the prior distribution of the states, *i.e.*, $\boldsymbol{\theta}_t^- \sim \mathcal{N}(\hat{\boldsymbol{\theta}}_t^-, \mathbf{R}_t)$ of Eq. 4, can be evaluated. We then define the probabilistic loss based on the prior to train OFlowNet.

5.1. Architecture

OFlowNet is composed of two components: the cost volume constructor and the flow estimator.

The cost volume constructor first extracts features from the two input images $\mathbf{I}_{t-1}$ and $\mathbf{I}_t$ respectively through seven 3×3 convolutions, three of which have a stride of 2 . The output feature maps $\mathbf{F}_{t-1}$ and $\mathbf{F}_t$ have a spatial size of one-eighth of the inputs and a channel number of c. Then, we build up a cost volume $\mathbf{C}_i \in \mathbb{R}_+^{w\times w\times c}$ for each pixel $\mathbf{p}_i$ of the feature map $\mathbf{F}_t$, so that

$$\mathbf{C}_i(\mathbf{o}) = \left|\frac{\mathbf{F}_t(\mathbf{p}_i)}{\|\mathbf{F}_t(\mathbf{p}_i)\|_2} - \frac{\mathbf{F}_{t-1}(\mathbf{p}_i+\mathbf{o})}{\|\mathbf{F}_{t-1}(\mathbf{p}_i+\mathbf{o})\|_2}\right|, \quad (11)$$

where w is the size of the search window which corresponds to $8w$ pixels in the full-resolution image, and $\mathbf{o} \in \{-w/2, ..., w/2\}^2$ is the spatial offset. We apply L2-normalization to the feature maps along the channel dimension before differentiation, as in [62, 34].

The following flow estimator operates over the cost volumes for flow inference. We use a U-Net with skip connections [44] as shown in Fig. 1, which first subsamples the cost volume by a factor of 8 for an enlarged receptive field and then upsamples it to the original resolution. The output is a $w \times w \times 1$ unbounded confidence map for each pixel. Related works usually attain flows by hard assignment based on the matching cost encapsulated by the cost volumes [62, 54]. However, it would cause non-differentiability in later steps where the optical flows are to be further used for spatial warping. Thus, we pass the confidence map through the differentiable *spatial softmax* operator [16] to compute the optical flow as the expectation of the pixel offsets inside the search window. Formally,

$$\hat{\mathbf{o}} \stackrel{\text{def}}{=} \mathrm{E}(\mathbf{o}) = \sum_{\mathbf{o}} \mathrm{softmax}(f_{\mathbf{o}}) \cdot \mathbf{o}, \quad (12)$$

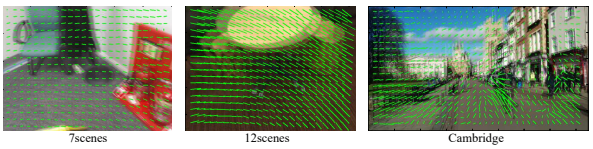


Figure 3: Sample optical flows predicted by OFlowNet over consecutive images (overlaid) of three different datasets [52, 57, 25].

where $f_{\mathbf{o}}$ is the confidence at offset $\mathbf{o}$. To fulfill the process noise modeling, *i.e.*, $\mathbf{w}_t$ in Eq. 3, we append three fully connected layers after the bottleneck of the U-Net to regress the logarithmic variance, as shown in Fig. 1. Sample optical flow predictions are visualized in Fig. 3.

5.2. Loss

Once the optical flows are computed, the state transition matrix $\mathbf{G}_t$ of Eq. 3 can be evaluated. We then complete the linear transition process of Eq. 3 by warping the scene coordinate map and uncertainty map from time $t-1$ towards t through bilinear warping [64]. Let $\hat{\boldsymbol{\theta}}^-_{(i)}$ and ${\sigma^-_{(i)}}^2$ be the warped scene coordinates and Gaussian variance, and ${w_{(i)}}^2$ be the Gaussian variance of the process noise of pixel $\mathbf{p}_i$ at time t. Then, the prior coordinates of $\mathbf{p}_i$, denoted as $\boldsymbol{\theta}^-_{(i)}$, should follow the distribution

$$\boldsymbol{\theta}^-_{(i)} \sim \mathcal{N}(\hat{\boldsymbol{\theta}}^-_{(i)}, {r_{(i)}}^2 \mathbb{I}_3), \tag{13}$$

where ${r_{(i)}}^2 = {\sigma^-_{(i)}}^2 + {w_{(i)}}^2$. Taking the negative logarithm of the PDF of $\boldsymbol{\theta}^-_{(i)}$, we get the loss of the process system as

$$\mathcal{L}_{prior} = \sum_{i=1}^{N} \left(3 \log r_{(i)} + \frac{\|\hat{\boldsymbol{\theta}}^-_{(i)} - \mathbf{y}_{(i)}\|_2^2}{2{r_{(i)}}^2} \right). \tag{14}$$

It is noteworthy that the loss definition uses the prior distribution of $\boldsymbol{\theta}^-_{(i)}$ to provide the weak supervision for training OFlowNet, with no recourse to the optical flow labeling.

One issue with the proposed process system is that it assumes no occurrence of occlusions or dynamic objects which are two outstanding challenges for tracking problems [27, 67]. Our process system partially addresses the issue by giving the uncertainty measurements of the process noise. As shown in Fig. 2(b), OFlowNet generally produces much larger uncertainty estimations for the pixels from occluded areas and dynamic objects. This helps to give lower weights to these pixels that have incorrect flow predictions in the loss computation.

6. The Filtering System

The measurement and process systems in the previous two sections have derived the likelihood and prior estimations of the scene coordinates $\boldsymbol{\theta}_t$, respectively. The filtering system aims to fuse both of them based on Eq. 9 to yield the posterior estimation.

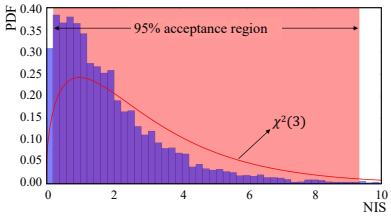


Figure 4: The illustration of NIS testing for the filtering system. The histogram draws the exemplar distribution of the Normalized Innovation Squared (NIS) values of the Kalman filter. The red curve denotes the PDF of the 3-DoF Chi-squared distribution $\chi^2(3)$. NIS testing works by filtering out the inconsistent predictions whose NIS values locate out of the 95% acceptance region (red shaded) of $\chi^2(3)$.

6.1. Loss

For a pixel $\mathbf{p}_i$ at time t, $\mathcal{N}(\mathbf{z}_{(i)}, {v_{(i)}}^2 \mathbb{I}_3)$ and $\mathcal{N}(\hat{\boldsymbol{\theta}}^-_{(i)}, {r_{(i)}}^2 \mathbb{I}_3)$ are respectively the likelihood and prior distributions of its scene coordinates. Putting the variables in Eqs. 6 & 9, we evaluate the innovation and the Kalman gain at pixel $\mathbf{p}_i$ as

$$\mathbf{e}_{(i)} = \mathbf{z}_{(i)} - \hat{\boldsymbol{\theta}}^-_{(i)}, \quad \text{and} \quad k_{(i)} = \frac{{r_{(i)}}^2}{{v_{(i)}}^2 + {r_{(i)}}^2}. \tag{15}$$

Imposing the linear Gaussian postulate of the Kalman filter, the fused scene coordinates of $\mathbf{p}_i$ with the least square error follow the posterior distribution below [37] :

$$\boldsymbol{\theta}^+_{(i)} \sim \mathcal{N}(\hat{\boldsymbol{\theta}}^+_{(i)}, {\sigma_{(i)}}^2 \mathbb{I}_3), \tag{16}$$

where $\hat{\boldsymbol{\theta}}^+_{(i)} = \hat{\boldsymbol{\theta}}^-_{(i)} + k_{(i)}\mathbf{e}_{(i)}$ and ${\sigma_{(i)}}^2 = {r_{(i)}}^2(1 - k_{(i)})$. Hence, the Kalman filtering system is parameter-free, with the loss defined based on the posterior distribution:

$$\mathcal{L}_{posterior} = \sum_{i=1}^{N} \left(3 \log \sigma_{(i)} + \frac{\|\hat{\boldsymbol{\theta}}^+_{(i)} - \mathbf{y}_{(i)}\|_2^2}{2{\sigma_{(i)}}^2} \right), \tag{17}$$

which is then added to the full loss that allows the end-to-end training of KFNet as below:

$$\mathcal{L}_{full} = \tau_1 \mathcal{L}_{likelihood} + \tau_2 \mathcal{L}_{prior} + \tau_3 \mathcal{L}_{posterior}. \tag{18}$$

6.2. Consistency Examination

In practice, the filter could behave incorrectly due to the outlier estimations caused by the erratic scene coordinate regression or a failure of flow tracking. This would induce accumulated state errors in the long run. Therefore, we use the statistical assessment tool, *Normalized Innovation Squared (NIS)* [4], to filter the inconsistent predictions during inference.

Normally, the innovation variable $\mathbf{e}_{(i)} \in \mathbb{R}^3$ follows the Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{S}_{(i)})$ as shown by Eq. 8, where

		One-shot Relocalization					Temporal Relocalization				
	scenes	MapNet [8]	CamNet [13]	Active Search [49]	DSAC++ [6]	SCoordNet (Ours)	VidLoc [11]	LSTM-KF [12]	VLocNet++ [43]	LSG [63]	KFNet (Ours)
7scenes	chess	0.08m, 3.25°	0.04m, 1.73°	0.04m, 1.96°	0.02m, 0.5°	0.019m, 0.63°	0.18m, -	0.33m, 6.9°	0.023m,1.44°	0.09m, 3.28°	0.018m, 0.65°
	fire	0.27m, 11.7°	0.03m, 1,74°	0.03m, 1.53°	0.02m, 0.9°	0.023m, 0.91°	0.26m, -	0.41m, 15.7°	0.018m, 1.39°	0.26m, 10.92°	0.023m, 0.90°
	heads	0.18m, 13.3°	0.05m, 1.98°	0.02m, 1.45°	0.01m, 0.8°	0.018m, 1.26°	0.21m, -	0.28m, 13.01°	0.016m, 0.99°	0.17m, 12.70°	0.014m, 0.82°
	office	0.17m, 5.15°	0.04m, 1.62°	0.09m, 3.61°	0.03m, 0.7°	0.026m, 0.73°	0.36m, -	0.43m, 7.65°	0.024m, 1.14°	0.18m, 5.45°	0.025m, 0.69°
	pumpkin	0.22m, 4.02°	0.04m, 1.64°	0.08m, 3.10°	0.04m, 1.1°	0.039m, 1.09°	0.31m, -	0.49m, 10.63°	0.024m, 1.45°	0.20m, 3.69°	0.037m, 1.02°
	redkitchen	0.23m, 4.93°	0.04m, 1.63°	0.07m, 3.37°	0.04m, 1.1°	0.039m, 1.18°	0.26m, -	0.57m, 8.53°	0.025m, 2.27°	0.23m, 4.92°	0.038m, 1.16°
	stairs	0.30m, 12.1°	0.04m, 1.51°	0.03m, 2.22°	0.09m, 2.6°	0.037m, 1.06°	0.14m, -	0.46m, 14.56°	0.021m,1.08°	0.23m, 11.3°	0.033m, 0.94°
	Average	0.207m, 7.78°	0.040m, 1.69°	0.051m, 2.46°	0.036m, 1.10°	**0.029m, 0.98°**	0.246m, -	0.424m, 11.00°	**0.022m**, 1.39°	0.190m, 7.47°	0.027m, **0.88°**
Cambridge	GreatCourt	-	-	-	0.40m, 0.2°	0.43m, 0.20°	-	-	-	-	0.42m, 0.21°
	KingsCollege	1.07m, 1.89°	-	0.42m, 0.55°	0.18m, 0.3°	0.16m, 0.29°	-	2.01m, 5.35°	-	-	0.16m, 0.27°
	OldHospital	1.94m, 3.91°	-	0.44m, 1.01°	0.20m, 0.3°	0.18m, 0.29°	-	2.35m, 5.05°	-	-	0.18m, 0.28°
	ShopFacade	1.49m, 4.22°	-	0.12m, 0.40°	0.06m, 0.3°	0.05m, 0.34°	-	1.63m, 6.89°	-	-	0.05m, 0.31°
	StMarysChurch	2.00m, 4.53°	-	0.19m, 0.54°	0.13m, 0.4°	0.12m, 0.36°	-	2.61m, 8.94°	-	-	0.12m, 0.35°
	Street	-	-	0.85m, 0.83°	-	-	-	3.05m, 5.62°	-	-	-
	Average [1]	1.63m, 3.64°	-	0.29m, 0.63°	0.14m, 0.33°	**0.13m, 0.32°**	-	2.15m, 6.56°	-	-	**0.13m, 0.30°**
DeepLoc		-	-	**0.010m, 0.04°**	-	0.083m, 0.45°	-	-	0.320m, 1.48°	-	**0.065m, 0.43°**

[1] The average does not include errors of *GreatCourt* and *Street* as some methods do not report results of the two scenes.

Table 2: The median translation and rotation errors of different relocalization methods. Best results are in bold.

One-shot			Temporal
DSAC++[6]	ESAC [7]	SCoordNet	KFNet
96.8%	97.8%	98.9%	**99.2%**

Table 3: The 5cm-5deg accuracy of one-shot and temporal relocalization methods on 12scenes [57].

$\mathbf{S}_{(i)} = ({v_{(i)}}^2 + {r_{(i)}}^2)\mathbb{I}_3$. Then, NIS $= \mathbf{e}_{(i)}^T \mathbf{S}_{(i)}^{-1} \mathbf{e}_{(i)}$ is supposed to follow the Chi-squared distribution with three degrees of freedom, denoted as $\chi^2(3)$. It is thus reasonable to see a pixel state as an outlier if its NIS value locates outside the acceptance region of $\chi^2(3)$. As illustrated in Fig. 4, we use the critical value of 0.05 in the NIS test, which means we have at least 95% statistical evidence to regard one pixel state as negative. The uncertainties of the pixels failing the test, *e.g.* $\sigma_{(i)}$, are reset to be infinitely large so that they will have no effect in later steps.

7. Experiments

7.1. Experiment Settings

Datasets. Following previous works [25, 5, 6, 43], we use two indoor datasets - *7scenes* [52] and *12scenes* [57], and two outdoor datasets - *DeepLoc* [43] and *Cambridge* [25] for evaluation. Each scene has been split into different strides of sequences for training and testing.

Data processing. Images are downsized to 640×480 for *7scenes* and *12scenes*, 848×480 for *DeepLoc* and *Cambridge*. The groundtruth scene coordinates of *7scenes* and *12scenes* are computed based on given camera poses and depth maps, whereas those of *DeepLoc* and *Cambridge* are rendered from surfaces reconstructed with training images.

Training. Our best practice chooses the parameter setting as $\tau_1 = 0.2$, $\tau_2 = 0.2$, $\tau_3 = 0.6$. The ADAM optimizer [26] is used with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We use an initial learning rate of $\gamma = 0.0001$ and then drop it with exponential decay. The training procedure has 3 stages. First, we train SCoordNet for each scene with the likelihood loss $\mathcal{L}_{likelihood}$ (Eq. 10). The iteration number is set to be proportional to the surface area of each scene and the learning rate drops from γ to $\gamma/2^5$. In particular, we use SCoordNet as the one-shot version of the proposed approach. Second, OFlowNet is trained using all the scenes for each dataset with the prior loss $\mathcal{L}_{prior}$ (Eq. 14). It also experiences the learning rate decaying from γ to $\gamma/2^5$. Each batch is composed of two consecutive frames. The window size of OFlowNet in the original images is set to 64, 128, 192 and 256 for the four datasets mentioned above, respectively, due to the increasing ego-motion through them. Third, we fine-tune all the parameters of KFNet jointly by optimizing the full loss $\mathcal{L}_{full}$ (Eq. 18) with a learning rate going from $\gamma/2^4$ to $\gamma/2^5$. Each batch in the third stage contains four consecutive frames.

7.2. Results

7.2.1 The Relocalization Accuracy

Following [5, 6, 11, 56], we use two accuracy metrics: (1) the median rotation and translation error of poses (see Table 2); (2) the 5cm-5deg accuracy (see Table 3), *i.e.*, the mean percentage of the poses with translation and rotation errors less than 5 cm and 5°, respectively. The uncertainty threshold λ (Sec. 3) is set to 5 cm for *7scenes* and *12scenes* and 50 cm for *DeepLoc* and *Cambridge*.

One-shot relocalization. Our SCoordNet achieves the lowest pose errors on *7scenes* and *Cambridge*, and the highest 5cm-5deg accuracy on *12scenes* among the one-shot methods, surpassing CamNet [13] and MapNet [8] which are the state-of-the-art relative and absolute pose regression methods, respectively. Particularly, SCoordNet outperforms the state-of-the-art structure-based methods DSAC++ [6] and ESAC [7], yet with fewer parameters (24M vs. 210M vs. 28M, respectively). The advantage of SCoordNet should be mainly attributed to the uncertainty modeling, as we will analyze in the supplementary material. It also surpasses Active Search (AS) [49] on *7scenes* and *Cambridge*, but underperforms AS on *DeepLoc*. We find that, in the experiments

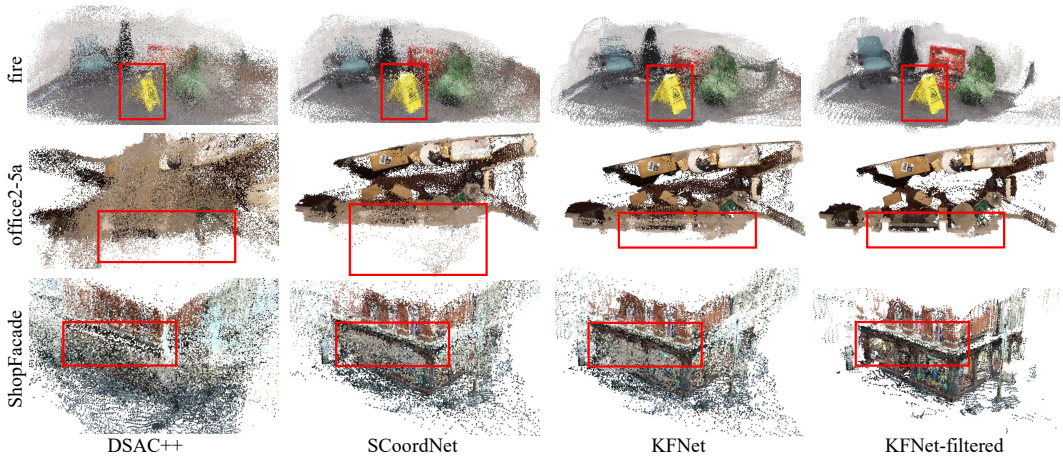


Figure 5: The point clouds predicted by different relocalization methods. Our SCoordNet and KFNet increasingly suppress the noise as highlighted by the red boxes and produce much neater point clouds than the state-of-the-art DSAC++ [6]. The KFNet-filtered panel filters out the points of KFNet of which the uncertainties are too large and gives rather clean and accurate mapping results.

	7scenes		12scenes		DeepLoc		Cambridge	
	mean	stddev	mean	stddev	mean	stddev	mean	stddev
DSAC++ [6]	28.8	33.1	28.8	47.1	-	-	467.3	883.7
SCoordNet	16.8	23.3	9.8	20.0	883.0	1520.8	272.7	497.6
KFNet	**15.3**	**21.7**	**7.3**	**13.7**	**200.79**	**398.8**	**241.5**	**441.7**

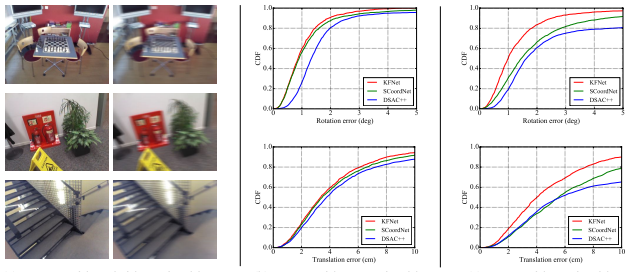
Table 4: The mean and standard deviation of predicted scene coordinate errors in centimeters.

of AS on *DeepLoc* [50], AS is tested on a SfM model built with both training and test images. This may explain why AS is surprisingly more accurate on *DeepLoc* than on other datasets, since the 2D-3D matches between test images and SfM tracks have been established and their geometry has been optimized during the SfM reconstruction.

Temporal relocalization. Our KFNet improves over SCoordNet on all the datasets as shown in Tables 2 & 3. The improvement on *Cambridge* is marginal as the images are over-sampled from videos sparsely. The too large motions between frames make it hard to model the temporal correlations. KFNet obtains much lower pose errors than other temporal methods, except that it has a larger translation error than VLocNet++ [43] on *7scenes*. However, the performance of VLocNet++ is inconsistent across different datasets. On *DeepLoc*, the dataset collected by the authors of VLocNet++, VLocNet++ has a much larger pose error than KFNet, even though it also integrates semantic segmentation into learning. The inconsistency is also observed in [50], which shows that VLocNet++ cannot substaintially exceed the accuracy of retrieval based methods [55, 2].

7.2.2 The Mapping Accuracy

Relocalization methods based on SCoRe [52, 6] can create a mapping result for each view by predicting per-pixel scene coordinates. Hence, relocalization and mapping can be seen as dual problems, as one can be easily resolved once the other is known. Here, we would like to evaluate the mapping accuracy with the mean and the standard deviation (stddev) of scene coordinate errors of the test images.



(a) Images without/with motion blur (b) CDFs without motion blur (c) CDFs with motion blur

Figure 6: (a) Artificial motion blur images. (b) & (c) The cumulative distribution functions (CDFs) of pose errors before and after motion blur is applied.

As shown in Table 4, the mapping accuracy is in accordance with the relocalization accuracy reported in Sec. 7.2.1. SCoordNet reduces the mean and stddev values greatly compared against DSAC++, and KFNet further reduces the mean error over SCoordNet by 8.9%, 25.5%, 77.3% and 11.4% on the four datasets, respectively. The improvements are also reflected in the predicted point clouds, as visualized in Fig. 5. SCoordNet and KFNet predict less noisy scene points with better temporal consistency compared with DSAC++. Additionally, we filter out the points of KFNet with uncertainties greater than λ as displayed in the KFNet-filtered panel of Fig. 5, which helps to give much neater and more accurate 3D point clouds.

7.2.3 Motion Blur Experiments

Although, in terms of the *mean scene coordinate error* in Table. 4, SCoordNet outperforms DSAC++ by over 41.6% and KFNet further improves SCoordNet by a range from 8.9% to 77.3%, the improvements in terms of the *median*

One-shot	Temporal			
SCoordNet	ConvLSTM [61]	TPooler [41]	SWeight [65]	KFNet
0.029m, 0.98°	0.040m, 1.12°	0.029m, 0.94°	0.029m, 0.95°	**0.027m, 0.88°**

Table 5: The median pose errors produced by different temporal aggregation methods on *7scenes*. Our KFNet achieves better pose accuracy than other temporal aggregation strategies.

pose error in Table 2 are not as significant. The main reason is that the RANSAC-based PnP solver diminishes the benefits brought by the scene coordinate improvements, since only a small subset of accurate scene coordinates selected by RANSAC matters in the pose accuracy. Therefore, to highlight the advantage of KFNet, we conduct more challenging experiments over motion blur images which are quite common in real scenarios. For the test image sequences of *7scenes*, we apply a motion blur filter with a kernel size of 30 pixels for every 10 images as shown in Fig. 6(a). In Fig. 6(b)&(c), we plot the cumulative distribution functions of the pose errors before and after applying motion blur. Thanks to the uncertainty reasoning, SCoordNet generally attains smaller pose errors than DSAC++ whether motion blur is present. While SCoordNet and DSAC++ show a performance drop after motion blur is applied, KFNet maintain the pose accuracy as shown in Fig. 6(b)&(c), leading to a more notable margin between KFNet and SCoordNet and demonstrating the benefit of the temporal modelling used by KFNet.

7.3. Ablation studies

Evaluation of Temporal Aggregation. This section studies the efficacy of our Kalman filter based framework in comparison with other popular temporal aggregation strategies including ConvLSTM [61, 28], temporal pooler (TPooler) [41] and similarity weighting (SWeight) [65, 66]. KFNet is more related to TPooler and SWeight which also use the flow-guided warping yet within an n-frame neighborhood. For equitable comparison, the same feature network and probabilistic losses as KFNet are applied to all. We use a kernel size of 8 for ConvLSTM to ensure a window size of 64 in images. The same OFlowNet structure and a 3-frame neighborhood are used for TPooler and SWeight for flow-guided warping.

Table 5 shows the comparative results on *7scenes*. ConvLSTM largely underperforms SCoordNet and other aggregation methods in pose accuracy, which manifests the necessity of explicitly determining the pixel associations between frames instead of implicit modeling. Although the flow-guided warping is employed, TPooler and SWeight only achieve marginal improvements over SCoordNet compared with KFNet, which justifies the advantage of the Kalman filtering system. Compared with TPooler and SWeight, the Kalman filter behaves as a more disciplined and non-heuristic approach to temporal aggregation that ensures an optimal solution of the linear Gaussian state-space model [17] defined in Sec. 3.

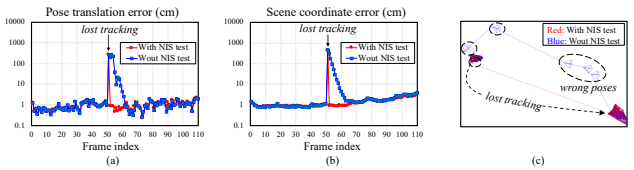


Figure 7: (a) & (b) With NIS testing [4], the errors of poses and scene coordinates quickly revert to normal after the lost tracking. (c) The poses of a sample sequence show that, without NIS testing, the lost tracking adversely affects the pose accuracy of the subsequent frames.

Evaluation of Consistency Examination Here, we explore the functionality of the consistency examination which uses NIS testing [4] (see Sec. 6.2). Due to the infrequent occurrence of extreme outlier predictions among the well-built relocalization datasets, we simulate the tracking lost situations by trimming a sub-sequence off each testing sequence of *7scenes* and *12scenes*. Let $\mathbf{I}_p$ and $\mathbf{I}_q$ denote the last frame before and the first frame after the trimming. The discontinuous motion from $\mathbf{I}_p$ to $\mathbf{I}_q$ would cause outlier scene coordinate predictions for $\mathbf{I}_q$ by KFNet. Fig. 7 plots the mean pose and scene coordinate errors of frames around $\mathbf{I}_q$ and visualizes the poses of a sample trimmed sequence. With the NIS test, the errors revert to a normal level promptly right after $\mathbf{I}_q$, whereas without the NIS test, the accuracy of poses after $\mathbf{I}_q$ is affected adversely. NIS testing stops the propagation of the outlier predictions of $\mathbf{I}_q$ into later steps by giving them infinitely large uncertainties, so that $\mathbf{I}_{q+1}$ will leave out the prior from $\mathbf{I}_q$ and reinitialize itself with the predictions of the measurement system.

8. Conclusion

This work addresses the temporal camera relocalization problem by proposing a recurrent network named KFNet. It extends the scene coordinate regression problem to the time domain for online pose determination. The architecture and the loss definition of KFNet are based on the Kalman filter, which allows a disciplined manner of aggregating the pixel-level predictions through time. The proposed approach yields the top accuracy among the state-of-the-art relocalization methods over multiple benchmarks. Although KFNet is only validated on the camera relocalization task, the immediate application alongside other tasks like video processing [20, 28] and segmentation [59, 39] , object tracking [34, 66] would be anticipated.

9. Acknowledgements

This work is supported by Hong Kong RGC GRF 16206819 & 16203518 and T22-603/15N.

References

[1] Theodore Wilbur Anderson. *An introduction to multivariate statistical analysis*, volume 2. 1958.

[2] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *CVPR*, 2016.

[3] Vassileios Balntas, Shuda Li, and Victor Prisacariu. Relocnet: Continuous metric learning relocalisation using neural nets. In *ECCV*, 2018.

[4] Yaakov Bar-Shalom, X Rong Li, and Thiagalingam Kirubarajan. *Estimation with applications to tracking and navigation: theory algorithms and software*. 2004.

[5] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. Dsac-differentiable ransac for camera localization. In *CVPR*, 2017.

[6] Eric Brachmann and Carsten Rother. Learning less is more-6d camera localization via 3d surface regression. In *CVPR*, 2018.

[7] Eric Brachmann and Carsten Rother. Expert sample consensus applied to camera re-localization. In *ICCV*, 2019.

[8] Samarth Brahmbhatt, Jinwei Gu, Kihwan Kim, James Hays, and Jan Kautz. Geometry-aware learning of maps for camera localization. In *CVPR*, 2018.

[9] Robert Castle, Georg Klein, and David W Murray. Video-rate localization in multiple maps for wearable augmented reality. In *ISWC*, 2008.

[10] Siddharth Choudhary and PJ Narayanan. Visibility probability structure from sfm datasets and applications. In *ECCV*, 2012.

[11] 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.

[12] Huseyin Coskun, Felix Achilles, Robert S DiPietro, Nassir Navab, and Federico Tombari. Long short-term memory kalman filters: Recurrent neural estimators for pose regularization. In *ICCV*, 2017.

[13] Mingyu Ding, Zhe Wang, Jiankai Sun, Jianping Shi, and Ping Luo. Camnet: Coarse-to-fine retrieval for camera relocalization. In *ICCV*, 2019.

[14] Chuong B Do. Gaussian processes. *Stanford University*, 2007.

[15] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. *IEEE robotics & automation magazine*, 13(2):99–110, 2006.

[16] Chelsea Finn, Xin Yu Tan, Yan Duan, Trevor Darrell, Sergey Levine, and Pieter Abbeel. Learning visual feature spaces for robotic manipulation with deep spatial autoencoders. 2015.

[17] Sylvia Frühwirth-Schnatter. Bayesian model discrimination and bayes factors for linear gaussian state space models. *Journal of the Royal Statistical Society*, 57(1):237–246, 1995.

[18] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. *PAMI*, 25(8):930–943, 2003.

[19] Mohinder S Grewal. Kalman filtering. In *International Encyclopedia of Statistical Science*, pages 705–708. 2011.

[20] Tae Hyun Kim, Mehdi S. M. Sajjadi, Michael Hirsch, and Bernhard Scholkopf. Spatio-temporal transformer network for video restoration. In *ECCV*, 2018.

[21] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*, 2017.

[22] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960.

[23] Alex Kendall and Roberto Cipolla. Modelling uncertainty in deep learning for camera relocalization. In *ICRA*, 2016.

[24] Alex Kendall and Roberto Cipolla. Geometric loss functions for camera pose regression with deep learning. In *CVPR*, 2017.

[25] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *ICCV*, 2015.

[26] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[27] Dieter Koller, Joseph Weber, and Jitendra Malik. Robust multiple car tracking with occlusion reasoning. In *ECCV*, 1994.

[28] Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. Learning blind video temporal consistency. In *ECCV*, 2018.

[29] Zakaria Laskar, Iaroslav Melekhov, Surya Kalia, and Juho Kannala. Camera relocalization by computing pairwise relative poses using convolutional neural network. In *ICCV*, 2017.

[30] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o (n) solution to the pnp problem. *IJCV*, 81(2):155, 2009.

[31] Xiaotian Li, Juha Ylioinas, Jakob Verbeek, and Juho Kannala. Scene coordinate regression with angle-based reprojection loss for camera relocalization. In *arXiv preprint arXiv:1808.04999*, 2018.

[32] Yunpeng Li, Noah Snavely, and Daniel P Huttenlocher. Location recognition using prioritized feature matching. In *ECCV*, 2010.

[33] Liu Liu, Hongdong Li, and Yuchao Dai. Efficient global 2d-3d matching for camera localization in a large-scale 3d map. In *ICCV*, 2017.

[34] Yongyi Lu, Cewu Lu, and Chi-Keung Tang. Online video object detection using association lstm. In *ICCV*, 2017.

[35] Yue Luo, Jimmy Ren, Zhouxia Wang, Wenxiu Sun, Jinshan Pan, Jianbo Liu, Jiahao Pang, and Liang Lin. Lstm pose machines. In *CVPR*, 2018.

[36] Daniela Massiceti, Alexander Krull, Eric Brachmann, Carsten Rother, and Philip HS Torr. Random forests versus neural networks—what's best for camera localization? In *ICRA*, 2017.

[37] Richard J Meinhold and Nozer D Singpurwalla. Understanding the kalman filter. *The American Statistician*, 37(2):123–127, 1983.

[38] Phuc Nguyen, Ting Liu, Gautam Prasad, and Bohyung Han. Weakly supervised action localization by sparse temporal pooling network. In *CVPR*, 2018.

[39] David Nilsson and Cristian Sminchisescu. Semantic video segmentation by gated recurrent flow propagation. In *CVPR*, 2018.

[40] David Novotny, Diane Larlus, and Andrea Vedaldi. Learning 3d object categories by looking around them. In *ICCV*, 2017.

[41] Tomas Pfister, James Charles, and Andrew Zisserman. Flowing convnets for human pose estimation in videos. In *ICCV*, 2015.

[42] Long Quan and Zhongdan Lan. Linear n-point camera pose determination. *PAMI*, 21(8):774–780, 1999.

[43] Noha Radwan, Abhinav Valada, and Wolfram Burgard. Vlocnet++: Deep multitask learning for semantic visual localization and odometry. *IEEE Robotics and Automation Letters*, 3(4), 2018.

[44] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 2015.

[45] Eric Royer, Maxime Lhuillier, Michel Dhome, and Jean-Marc Lavest. Monocular vision for mobile robot localization and autonomous navigation. *IJCV*, 74(3):237–260, 2007.

[46] Soham Saha, Girish Varma, and CV Jawahar. Improved visual relocalization by discovering anchor points. In *BMVC*, 2018.

[47] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *CVPR*, 2019.

[48] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Fast image-based localization using direct 2d-to-3d matching. In *ICCV*, 2011.

[49] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Efficient & effective prioritized matching for large-scale image-based localization. *PAMI*, (9):1744–1756, 2017.

[50] Torsten Sattler, Qunjie Zhou, Marc Pollefeys, and Laura Leal-Taixe. Understanding the limitations of cnn-based absolute camera pose regression. In *CVPR*, 2019.

[51] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

[52] 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.

[53] Jie Song, Limin Wang, Luc Van Gool, and Otmar Hilliges. Thin-slicing network: A deep structured model for pose estimation in videos. In *CVPR*, 2017.

[54] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *CVPR*, 2018.

[55] Akihiko Torii, Relja Arandjelovic, Josef Sivic, Masatoshi Okutomi, and Tomas Pajdla. 24/7 place recognition by view synthesis. In *CVPR*, 2015.

[56] Abhinav Valada, Noha Radwan, and Wolfram Burgard. Deep auxiliary learning for visual localization and odometry. In *ICRA*, 2018.

[57] Julien Valentin, Angela Dai, Matthias Nießner, Pushmeet Kohli, Philip Torr, Shahram Izadi, and Cem Keskin. Learning to navigate the energy landscape. In *3DV*, 2016.

[58] Julien Valentin, Matthias Nießner, Jamie Shotton, Andrew Fitzgibbon, Shahram Izadi, and Philip HS Torr. Exploiting uncertainty in regression forests for accurate camera relocalization. In *CVPR*, 2015.

[59] Sepehr Valipour, Mennatullah Siam, Martin Jagersand, and Nilanjan Ray. Recurrent fully convolutional networks for video segmentation. In *WACV*, 2017.

[60] 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.

[61] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *NIPS*, 2015.

[62] Jia Xu, Rene Ranftl, and Vladlen Koltun. Accurate optical flow via direct cost volume processing. In *CVPR*, 2017.

[63] Fei Xue, Xin Wang, Zike Yan, Qiuyuan Wang, Junqiu Wang, and Hongbin Zha. Local supports global: Deep camera relocalization with sequence enhancement. In *ICCV*, 2019.

[64] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, 2017.

[65] Xizhou Zhu, Yujie Wang, Jifeng Dai, Lu Yuan, and Yichen Wei. Flow-guided feature aggregation for video object detection. In *ICCV*, 2017.

[66] Zheng Zhu, Wei Wu, Wei Zou, and Junjie Yan. End-to-end flow correlation tracking with spatial-temporal attention. In *CVPR*, 2018.

[67] Danping Zou and Ping Tan. Coslam: Collaborative visual slam in dynamic environments. *PAMI*, 35(2), 2013.