# VidLoc: A Deep Spatio-Temporal Model for 6-DoF Video-Clip Relocalization

Ronald Clark[1], Sen Wang[1], Andrew Markham[1], Niki Trigoni[1], Hongkai Wen[2]
[1]University of Oxford
[2]University of Warwick
firstname.lastname@cs.ox.ac.uk

## Abstract

*Machine learning techniques, namely convolutional neural networks (CNN) and regression forests, have recently shown great promise in performing 6-DoF localization of monocular images. However, in most cases image-sequences, rather only single images, are readily available. To this extent, none of the proposed learning-based approaches exploit the valuable constraint of temporal smoothness, often leading to situations where the per-frame error is larger than the camera motion. In this paper we propose a recurrent model for performing 6-DoF localization of video-clips. We find that, even by considering only short sequences (20 frames), the pose estimates are smoothed and the localization error can be drastically reduced. Finally, we consider means of obtaining probabilistic pose estimates from our model. We evaluate our method on openly-available real-world autonomous driving and indoor localization datasets.*

## 1. Introduction

Localization of monocular images is a fundamental problem in computer vision and robotics. Camera localization forms the basis of many functions in computer vision where it is an important component of the Simultaneous Localization and Mapping (SLAM) process and has direct application, for example, in the navigation of autonomous robots and drones in first-response scenarios or the localization of wearable devices in assistive living applications.

The most common means of performing 6-DOF pose estimation using visual data is to make use of specially-built models, which are constructed from a vast number of local features that have been extracted from the images captured during mapping. The 3D locations of these features are then found using a Structure-from-Motion (SfM) process, creating a many-to-one mapping from feature descriptors to 3D points. Traditionally, localizing a new query image against these models involves finding a large set of putative correspondences. The pose is then found using RANSAC to reject outlier correspondences and optimize the camera pose on inliers. Although this traditional approach has proven to be incredibly accurate in many situations, it faces numerous and significant challenges. These methods rely on local and unintuitive hand-crafted features, such as SIFT keypoints. Because of their local nature, establishing a sufficient number of reliable correspondences between the image pixels and the map is very challenging. Spurious correspondences arise due to both "well-behaved" phenomena such as sensor noise and quantization effects as well as pure outliers which arise due to the local correspondence assumptions not being satisfied [6]. These include inevitable environmental appearance changes due to, for example, changing light levels or dynamic elements such as clutter or people in the frame or the opening and closing of doors. These aspects conspire to give rise to a vast number of suprious correspondences, making it difficult to use for any purpose but the localization of crisp and high-resolution images. Secondly, the maps often consists of millions of elements which need to be searched, making it very computationally intensive and difficult to establish correspondences in real-time.

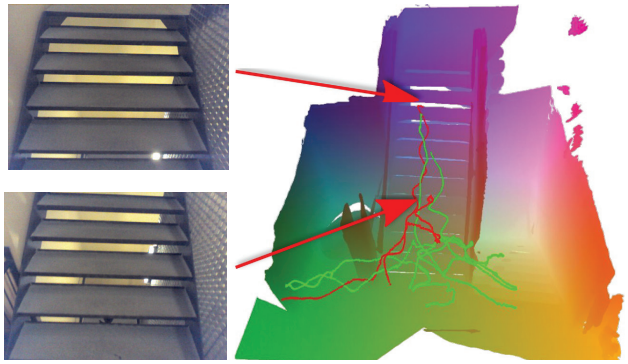


Figure 1: An extreme example of perceptual aliasing in the Stairs scene of the Microsoft 7-Scenes dataset. One of the frames is taken at the bottom of the staircase and the other near the top. Using only single frames, as in the competing approaches, it would be impossible to correctly localize these images.

1063-6919/17 $31.00 © 2017 IEEE
DOI 10.1109/CVPR.2017.284

Recently, however, it has been shown that machine learning methods such as random forests [20] and convolutional neural networks (CNNs) [10] have the ability to act as a regression model which directly estimates pose from an input image with no expensive feature extraction or feature matching processes required. These methods consider the input images as being entirely uncorrelated and produce independent pose estimates that are incredibly noisy when applied to image sequences. On most platforms, including smart-phones, mobile robots and drones, image-sequences are readily obtained and have the potential to greatly enhance the accuracy of these approaches and promising results have been obtained for sequence-based learning for relative pose estimation [4]. Therefore, in this paper we consider ways in which we can leverage the temporal dependencies in image-sequences to improve the accuracy of 6-DoF camera re-localization. Furthermore, we show how we can in essence unify map-matching, model-based localization, and temporal filtering all in one, extremely compact model.

### 1.1. Related Work

**Map-matching** Map matching methods make use of a map of a space either in the form of roads and traversable paths or a floor-plan of navigable and non-navigable areas to localize a robot as it traverses the environment. Map-matching techniques are typified by their non-reliance on strict data-association and can use both exteroceptive (eg. laser scans) or interoceptive (odometry, the trajectory or the motion of the platform) sensors to obtain a global pose estimate. The global pose estimate is obtained through probabilistic methods such as sequential Monte Carlo (sMC) filtering [7] or hidden Markov models (HMMs) [15]. These methods inherently incorporate sequential observations, but accuracy is inferior to localizing against specialised maps, such as a 3D map of sparse features.

**Sparse feature based localization** When a 3D model of discriminative feature points is available (eg. obtained using SfM) then the poses of query images can be found using camera re-sectioning. Matching against large 3D models is generally very computationally expensive and requires lots of memory space to store the map. A number of approaches have been proposed to improve the efficiency of standard 3D-to-2D feature matching between the image and the 3D model [24]. For example, [16] propose a quantized feature vocabulary for direct 2D-to-3D matching with the camera pose being found using RANSAC in combination with a PnP algorithm and in [17] an active search method is proposed to efficiently find more reliable correspondences. [13] propose a client-server architecture where the client exploits sequential images to perform high-rate local 6-DoF tracking which is then combined with lower-rate global localization updates from the server, entirely eliminating the need for loop-closure. The authors propose various methods to integrate the smooth local poses with the global updates. In [11] the authors consider means of improving the global accuracy by introducing temporal constraints into the image registration process by regularizing the poses trough smoothing.

**Scene coordinate regression forests** of Shotton et al. [20] use a regression forest to learn the mapping between the pixels of an RGB-D input image and the scene co-ordinates of a previously established model. In essence the regression forest learns the function $f : (r, g, b, d, u, v) \rightarrow (U, V, W)$. To perform localization, a number of RGB-D pixels from the query image are fed through the forest and a RANSAC-based pose computation is used to determine a consistent and accurate final camera pose. To account for the temporal regularity of image sequences, the authors consider a frame-to-frame extension of their method. To accomplish this, they initialize one of the pose hypotheses with that obtained from the previous frame, which results in a significant improvement in localization accuracy. Although extremely accurate, the main disadvantage of this approach is that it requires depth images to function and does not eliminate the expensive RANSAC procedure.

**CNN features** Deep learning is quickly becoming the dominant approach in computer vision. The many layers of a pre-trained CNN form a hierarchical model with increasingly higher level representations of the input data as one moves up the layers. It has been shown that many computer vision related tasks benefit from using the output from these upper layers as feature representations of the input images. These features have the advantage of being low-level enough to provide representations for a large number of concepts, yet are abstract enough to allow these concepts to be recognized using simple linear classifiers [19]. They have shown great success applied to a wide range of tasks including logo classification [1], and more closes related to our goals, scene recognition [25] and place recognition [22].

**Posenet** [10] demonstrated the feasibility of estimating the pose of a single RGB image by using a deep CNN model to regress directly on the pose. For practical camera relocalization, Posenet is far from ideal. For example, on the Microsoft 7-Scenes dataset it achieves a $0.48m$ error where the model space is only $2.5m \times 1m \times 1m$. Our approach enhances the localization accuracy by incorporating a temporal aspect in the model, producing smoother and more accurate estimates.

### 1.2. Contributions

In this paper, we propose a recurrent model for reducing the pose estimation error by using multiple frames for the pose prediction. Our specific contributions are as follows:

1. We present a deep spatio-temporal model for efficent global localization from a monocular image sequence.

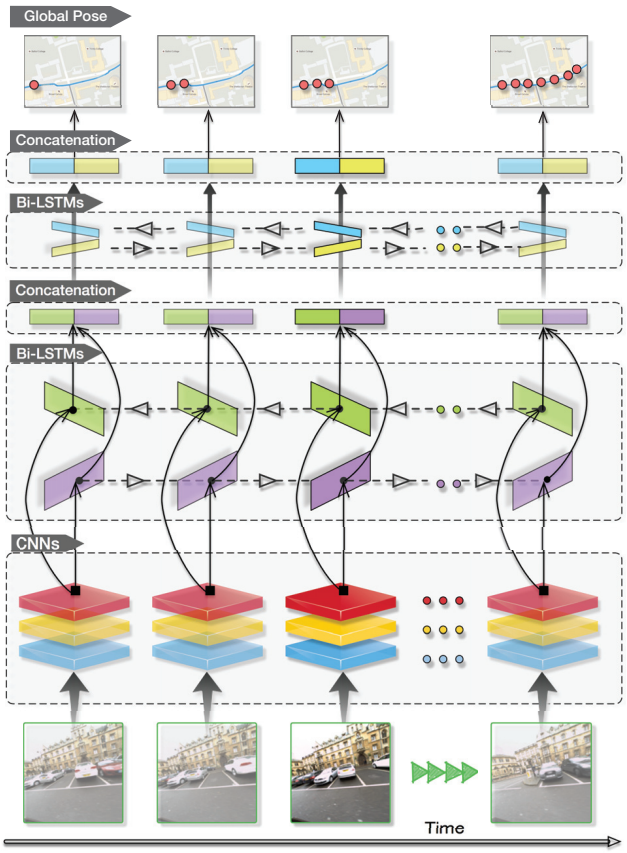



Figure 2: The CNN-RNN network for video-clip localization.

2. We integrate into our network a method for obtaining the instantaneous covariances of pose estimates.

3. We evaluate our approach to two large open datasets and show that the proposed spatio-temporal model performs significantly outperforms a smoothing baseline.

## 2. Proposed Model

In this section we outline our proposed model for video-clip localization, VidLoc, a high-level overview of which is shown in Figure 2. Our model processes the video image frames using CNN and integrates temporal information through a bidirectional LSTM.

### 2.1. Image Features: CNN

The goal of the CNN part of our model is to extract relevant features from the input images that can be used to predict the global pose of an image. A CNN consists of stacked layers performing convolution and pooling operations on the input image. There are a large number of CNN architectures that have been proposed, most for classifying objects in images and trained on the Imagenet database. These models, however, generalize well to other tasks, including pose estimation. As in the Posenet [10] paper, VGGNet [21] is able to produce more accurate pose estimates, but incurs a high-computational cost due to its very deep architecture. As we are interested in processing multiple images in a temporal sequence we adopt the GoogleNet Inception [23] architecture for the VidLoc CNN. We use only the convolutional and pooling layers of GoogleNet and drop all the fully-connected layers. In our experiments, we explore the impact on computational efficiency incurred vs. the increase in accuracy obtained using multiple frames.

### 2.2. Temporal Modelling: Bidirectional RNN

In Posenet and many other traditional image based localization approaches, the pose estimates are produced entirely independently for each frame. However, when using image-streams with temporal continuity, a great deal of pose information can be gained by exploiting the temporal dependencies. For example, adjacent images often contain views of the same object which can boost the confidence in a particular location, and there are also tight constraints on the motion that can be undergone in-between frames - a set of frames estimated to be at a particular location are very unlikely to contain one or two located far away.

To capture these dynamic dependencies, we make use of the LSTM model in our network. The LSTM [8] extends standard RNNs to enable them to learn long-term time dependencies. This is accomplished by including a *forget gate*, input and output *reset gates* and a *memory cell*. The flow of information into and out-of the memory cell is regulated by the forget and input gates. This allows the network to overcome the vanishing gradient problem during training and thereby allow it to learn long-term dependencies. The input to the LSTM is the output of the CNN consisting of a sequence of feature vectors, $\mathbf{x}_t$. The LSTM maps the input sequence to the output sequence consisting of the global pose parameterised as a 7-dimensional vector, $\mathbf{y}_t$ consisting of a translation vector and orientation quaternion. The activations of the LSTM are computed by interatively applying the following operations on each timestep

$$\begin{aligned} f_t &= \sigma_g(W_f\mathbf{x}_t + U_f\mathbf{h}_{t-1} + b_f) \\ i_t &= \sigma_g(W_i\mathbf{x}_t + U_i\mathbf{h}_{t-1} + b_i) \\ o_t &= \sigma_g(W_o\mathbf{x}_t + U_o\mathbf{h}_{t-1} + b_o) \\ c_t &= f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c\mathbf{x}_t + U_c\mathbf{h}_{t-1} + b_c) \\ \mathbf{h}_t &= o_t \circ \sigma_h(c_t) \\ \mathbf{y}_t &= \sigma_o(W_y h_t + b_y) \end{aligned} \tag{1}$$

where $W, U$ and $b$ are the parameters of the LSTM, $f_t, i_t, o_t$ are the gate vectors, $\sigma_g$ is the non-linear activation function and $h_t$ is the hidden activation of the LSTM. For the inner activations, we use a hyperbolic tangent function and for the output $\sigma_o$ we use a linear activation. A limitation of

the standard LSTM model is that it is only able to make use previous context in predicting the current output. For our monocular image-sequence pose prediction application we have a sliding window of frames available at any one instance in time and thus we can exploit both future and past contextual information predicting the poses for each frame in the sequence. For this reason, we adopt a Bidirectional architecture [18] for our LSTM model. The bidirectional model assumes the same state equations as in 1, but uses both future and past information for each frame by using two hidden states, $\overleftarrow{\mathbf{h}}_t$ and $\overrightarrow{\mathbf{h}}_t$, one for processing the data forwards and and the other for processing backwards, as shown in Figure 3. The hidden states are then combined to form a single hidden state $\mathbf{h}_t$ through a concatenation operation

$$\mathbf{h}_t = \left[\overleftarrow{\mathbf{h}}_t, \overrightarrow{\mathbf{h}}_t\right] \tag{2}$$

The output pose is computed from this hidden layer as in 1.

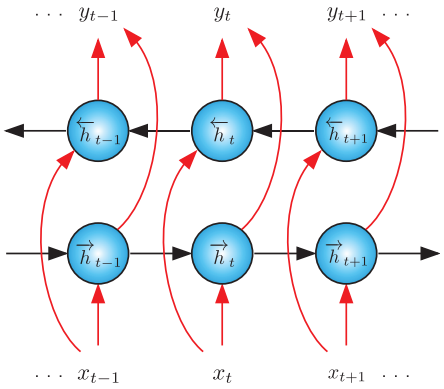



Figure 3: The structure of a bidirectional RNN [18].

### 2.3. Network Loss

In order to train the network we use the sum of the Euclidean error magnitude of both the translation and orientation. To compute the loss, we separate the output of the LSTM into the translation $\mathbf{x}_t$ and orientation $\mathbf{q}_t$

$$\mathbf{y}_t = [\mathbf{x}_t, \mathbf{q}_t] \tag{3}$$

and use a weighted sum of the error magnitudes of the two component vectors

$$\mathcal{L} = \sum_{t=1}^{T} \alpha_1 ||\mathbf{x}_t - \hat{\mathbf{x}}_t|| + \alpha_2 ||\mathbf{q}_t - \hat{\mathbf{q}}_t|| \tag{4}$$

We propagate the loss through the temporal frames in each training sequence by unrolling the network and performing back-propagation through time. To update the weights of the layers, we make use of the Adam optimizer.

### 2.4. Probabilistic Pose Estimates

Pose estimation methods, no matter how accurate, will always be subject to a degree of uncertainty. Being able to correctly model and predict uncertainty is thus a key component of any useful visual localization method. The euclidean sum-of-squares error which we defined in Sec. 2.3 results in a network which approximates only the uni-modal conditional mean of the pose as defined by the training data. In essence the output of the network can be regarded as predicting $\mu_{\mathbf{x}}$, the mean of the conditional pose distribution $p([\mathbf{x}, \mathbf{q}]|I) = \mathcal{N}(\mu_{[\mathbf{x},\mathbf{q}]}, \sigma)$ where the Guassian assumption is induced by the use of the square error loss. For the unlikely case where the actual posterior pose distribution is Gaussian, this mean represents the optimal distribution in a maximum-likelihood sense. However, for global camera re-localization as we are concerned with in this paper, this assumption is unlikely. In many instances the appearance of a space is similar at multiple locations, for example, two corridors in a building may appear very similar (known as the "perceptual aliasing" problem and in most instances cannot be addressed using visual data alone).

In [9], one possible means of representing multi-modal uncertainty in the global pose estimation was considered. In this work, the authors create a Bayesian convolutional neural network by using dropout as a means of sampling the model weights. The posterior distribution of the model weights $p(\mathbf{W}|\mathbf{X}, \mathbf{Y})$ is intractable and they use variational inference to approximate it as proposed in [5]. To produce probabilistic pose estimates, Monte Carlo pose samples are drawn and the mean and variance determined from these. Although this models the uncertainty in the *model weights* correctly (i.e. the distribution of the model weights according to the training data), it does not fully capture the uncertainty of the pose estimates.

To model the pose uncertainty, we adopt the mixture density networks method [2]. This approach replaces the Gaussian with a mixture model, allowing a multi-modal posterior output distribution to be modelled. Using this approach, the pose estimates now take the form

$$p([\mathbf{x}, \mathbf{q}]|I) = \sum_{i=1}^{M} \alpha_i(I) \mathcal{N}_i\left(\mu_{[\mathbf{x},\mathbf{q}]}(I), \sigma(I)\right) \tag{5}$$

where $\mathcal{N}_i\left(\mu_{[\mathbf{x},\mathbf{q}]}, \sigma|I\right)$ is a mixture component and $\alpha_i$ are the coefficients of the mixture distribution which satisfy the constraint $\sum_i \alpha_i = 1$. The mixing components are a function of the input image which is modelled by the network. As in the single Gaussian case, the network is trained to maximize the likelihood of the training data.

## 3. Experiments

In this section, the proposed approach is evaluated on outdoor and indoor datasets by comparing with the state-of-the-art methods.

### 3.1. Datasets

Two well-known public datasets are employed in our experiments. They demonstrate indoor human motion and outdoor autonomous car driving scenarios, respectively.

The first is the Microsoft 7-Scenes Dataset which contains RGB-D image sequences of 7 different indoor environments [20], created by using a Kinect sensor. It has been widely used for camera tracking and relocalization [10]. The images were captured at $640 \times 480$ resolution with ground truth from KinectFusion system. Since there are several image sequences of one scene and each sequence is composed of about 500-1000 image frames, it is ideal for our experiments. Ground truth camera poses for the dataset are obtained using the KinectFusion algorithm [14] to produce smooth camera tracks and a dense 3D model of each scene. In our experiments, all the 7 scenes are adopted to evaluate the proposed method. We use the same Train and Test split of the sequences as used in the original paper. This dataset consists of both RGB and depth images. Although we focus mainly on RGB-only localization, our method extends naturally to the RGB-D case.

In order to further test the performance in large-scale outdoor environments, the recently released Oxford RobotCar dataset [12] is used. It was recorded by using an autonomous Nissan LEAF car traversing in the central Oxford for a year period. The dataset contains high-resolution images from a Bumblebee stereo camera, LiDAR scanning, and GPS/INS. Since different weather conditions, such as sunny and snowy days, are exhibited in the dataset, it is very challenging for some tasks based on vision, e.g., global localization and loop closure detection across long terms and seasons. Because global re-localization does not need to have high-frequency images, the frame rate is about 1Hz in our robotcar experiments.

### 3.2. Competing algorithms

In this section we describe the experiments that we performed on the Microsoft 7-Scenes dataset. We compare our approach to the current state-of-the-art monocular camera localization methods.

**Smoothing baseline** The traditional means of integrating temporal information is to perform a filtering or smoothing operation on the independent pose predictions for each frame. We thus compare our method to a smoothing operation in order to investigate the advantage of using an RNN to capture the temporal information and whether global pose accuracies obtained for each frame are indeed more accurate than independent pose predictions. For our smoothing baseline, we use the spline fitting approach as per [11].

**Posenet** Posenet uses a CNN to predict the pose of an input RGB image. The Posenet network is the GoogleNet architecture with the top-most fully connected layer removed and replaced by one with a 7-dimensional output and trained to predict the pose of the image.

**Score-Forest** The Score-Forest [20] approach trains a random regression forest to predict the scene coordinates of pixels in the images. A set of predicted scene coordinates is then used to determine the camera pose using a RANSAC-loop. We use the open source implementation for our experiments[1].

**Additional comparisons** Comparison to [16]. For RobotCar (Fig **??**) we extract SURF features and assign 3D locations using LiDAR data. We also provide a comparison on 7 Scenes to [3] from the results as presented in [3].

### 3.3. Experiments on Microsoft 7-Scenes Dataset

The results of our experiments testing the accuracy of our method are shown in Table 1. The proposed method significantly outperforms the Posenet approach in all of the test scenes, resulting in a $23.4\% - 55\%$ increase in accuracy. The SCoRe-forest outperforms the the RGB-only VidLoc. However, this is strictly not a fair comparison for two reasons: firstly, SCoRe-forest requires depth images as input; secondly, the SCoRe-forest sometimes produces pose estimates with gross errors although these are rejected by the RANSAC-loop, which means that pose estimates are not available for all frames. In contrast, our method produces reliable estimates for the entire sequence.

We tested our method using both depth and RGB input and although our method seamlessly utilises the depth images when available, a disadvantage is that it cannot utilise the depth information to the extent that the SCoRe-Forest is able to. This is evidenced in the accuracy results reported in Table 1 where it can be seen that although our method consistently achieves centimeter accuracy, it does not outperform the SCoRe-Forest. This is surprising but perhaps indicative of the operation of the network. This suggests that the network learns to perform pose prediction in a similar fashion to an appearance based localization method. In this manner, it uses both the RGB and the depth information in the same way. This is in contrast to the SCoRe-forest approach where the depth information is explicitly used in a geometric pose computation by means of the PnP algorithm. We note, however, that our method still has the advantage of being able to operate on RGB data when no depth information is available and is able to produce global pose estimates for all frames whereas the SCoRe-forest cannot.

[1] https://github.com/ISUE/relocforests

Table 1: Comparison to state-of-the-art approaches to monocular camera localization

| Scene | Frames | | Spatial | Score | Posenet | Bayesian | Smoothing | VidLoc | VidLoc | VidLoc |
|---|---|---|---|---|---|---|---|---|---|---|
| | Train | Test | Extent | Forest | | Posenet | Baseline | | RGB-D | Depth |
| Chess | 4000 | 2000 | 3x2x1m | 0.03m | 0.32m | 0.37m | 0.32m | 0.18m | 0.16m | 0.19m |
| Office | 6000 | 4000 | 2.5x2x1.5m | 0.04m | 0.48m | 0.48m | 0.38m | 0.26m | 0.24m | 0.32m |
| Fire | 2000 | 2000 | 2.5x1x1m | 0.05m | 0.47m | 0.43m | 0.45m | 0.21m | 0.19m | 0.22m |
| Pumpkin | 4000 | 2000 | 2.5x2x1m | 0.04m | 0.47m | 0.61m | 0.42m | 0.36m | 0.33m | 0.15m |
| Red kitchen | 7000 | 5000 | 4x3x1.5m | 0.04m | 0.59m | 0.58m | 0.57m | 0.31m | 0.28m | 0.38m |
| Stairs | 2000 | 1000 | 2.5x2x1.5m | 0.32m | 0.47m | 0.48m | 0.44m | 0.26m | 0.24m | 0.27m |
| Heads | 1000 | 1000 | 2x0.5x1m | 0.06m | 0.29m | 0.31m | 0.19m | 0.14m | 0.13m | 0.27m |
| Average | | | | | | | | | | |

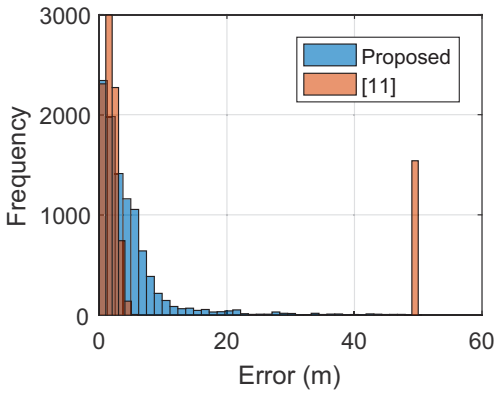



Figure 4: Error histogram of VidLoc compared to a sparse-feature based method [16] on the RobotCar dataset.

| | $5cm, 5°$ | Avg. Error |
|---|---|---|
| 1 | 40.7% | - |
| 2 | - | 46.9cm, 5.4° |
| 3 | - | 25.7cm, 3.8° |
| 4 | 55.2% | 6.1cm, 2.7° |
| 5 | N/A | 6cm, 2.89° |

Table 2: Additional comparisons on RobotCar and 7 Scenes. (1) Sparse RGB, (2) PoseNet, (3) Proposed, (4) Brachmann et al., (5) Sattler et al. [15] .

**Effect of sequence length** A key result of this paper is shown in Figure 5 which depicts the localization error as a function of the sequence length used. We have trained the models using sequence lengths of 200 frames in order to test the ability of the model to generalize to longer sequences. In all cases we ensure that the error is averaged over the same number and an even distribution across the test sequence. As expected, increasing the number of frames improves the localization accuracy. We also see that the model is able to generalize to longer sequence (i.e we still get an improvement in accuracy for sequence lengths greater than 200). At very long sequence lengths we experience diminishing returns - however this is not necessarily a product of the models inability to use this data but rather the actual utility of very long-term dependencies in predicting the current pose.

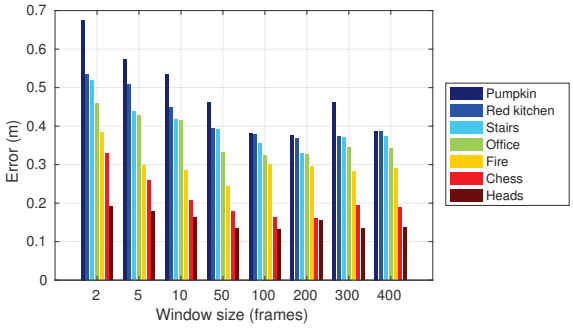



Figure 5: The effect of window length on pose accuracy for the sequences in the Microsoft 7-Scenes dataset.

**Timings** Our approach improves on the accuracy of Posenet, yet has very little impact on the computational time. This is because processing each frame only relies on the hidden state of the RNN from the previous time instance and image data of the current frame. Predicting a pose thus only requires a forward pass of the image through the CNN and propagating the hidden state. On our test machine with a Titan X Pascal GPU, this takes only 18ms using GoogleNet and 43ms using a VGG16 CNN. An interesting observation from our experiments is that the training time to create a usable localization network using the fine-tuning approach with Imagenet initialization is actually rather short. Typically convergence time (to around 90%) of final accuracy on the test data is around 50s.

**Uncertainty output** The 7-Scenes indoor dataset is extremely challenging, mainly due to the problem of perceptual aliasing as shown in Figure 1. One image was taken from the bottom of the staircase while the other was taken near the top. For comparing the uncertainty to [9] we use the Bayesian PoseNet implementation provided by the authors

of [9]. Figure 6 shows a visualization of the predicted uncertainty and the actual error in the format of [9]. The percentage of pose errors fall within the $3\sigma$ bound for the proposed adopted uncertainty method is $97.2\%$ and [9] is $98.1\%$ (this value is ideally $99.7\%$). Both approaches produce high-quality uncertainty estimates, although the proposed is a bit less conservative. The proposed requires no approximation or sampling. From the figure it is evident that the predicted distribution adaquately. However, in many cases we found that the predicted variance is rather high and we leave it as future work to improve the variance prediction.

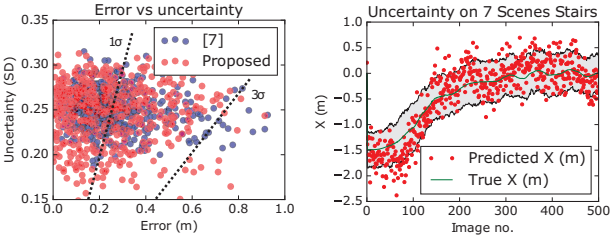



Figure 6: (a)Comparison of uncertainty to [7] and (b) visualization of proposed uncertainty prediction ($1\sigma$) and trajectory.

### 3.4. Experiments on RobotCar Dataset

The experiments on the Oxford RobotCar Dataset are given in this section. Since the GPS/INS poses are relatively noisy (zig-zag track), they are fused with stereo visual odometry by using pose graph SLAM to produce smooth ground truth for training. In our experiments, three image sequences are used for training, while the trained models are tested on another new testing sequence.

(a) Images on a same location at different times.

(b) Images on different locations but close times.

Figure 7: Images of the RobotCar dataset to show limited appearance distinction with dynamic changes on a same location and perceptual aliasing among different locations.

The image sequences selected are very challenging for global re-localization. As shown in Figure 7, the images are mostly filled with roads and trees, which do not have distinct and consistent appearance features. Specifically, three images of a same location yet captured at different times are presented in Figure 7a. Although they are taken at a same position, the cars parking along the road introduce significant appearance changes. Without viewing the buildings around, the only consistent objects which can be useful for global re-localization are the trees and roads. However, they are subtle in terms of image context. For example, Figure 7b shows sample images of three different locations which share very similar appearance. Again, this perceptual aliasing makes global re-localization more challenging by only using one single image.

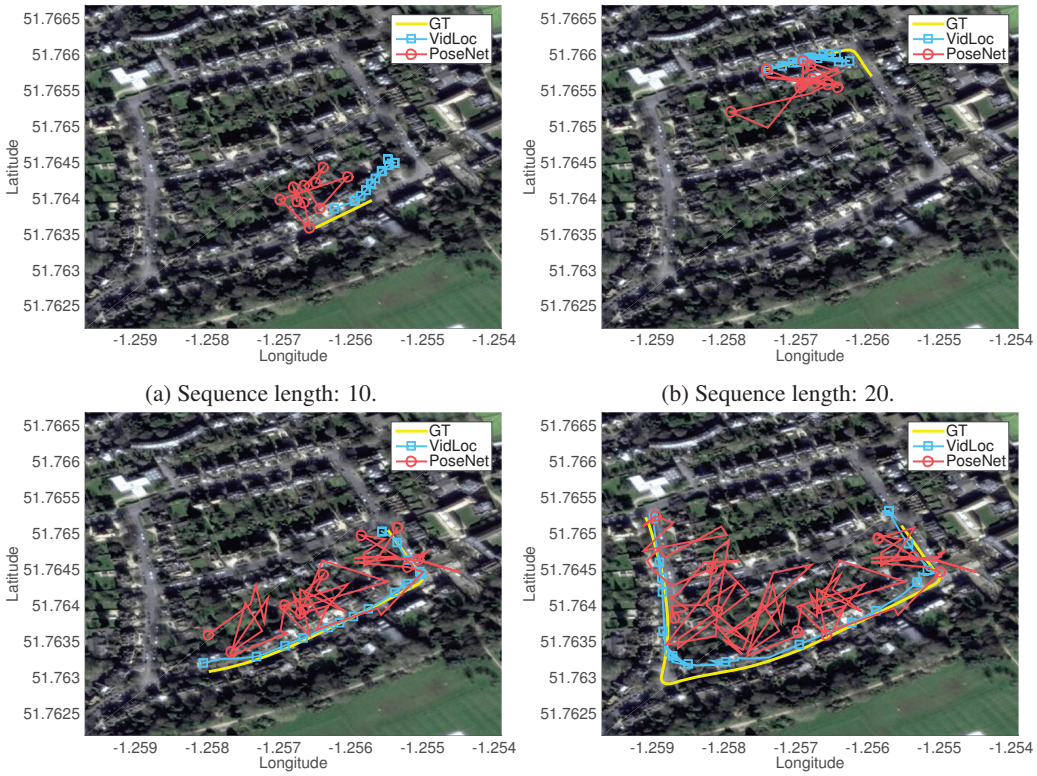
The global re-localization results of the testing image sequence with lengths 10, 20, 50 and 100 are shown in Figure 8 against ground truth. They are also superimposed on Google Map. It can be seen that the result of the proposed method improves as the length of the sequence increases, and the re-localization results of the lengths 50 and 100 match with the roads consistently. It is interesting to see that its trajectories are also able to track the shape of motion by end-to-end learning. In contrast, the Posenet which uses a single image suffers from noisy pose estimates around the ground truth. This experiment validates the effectiveness and necessary of using sequential images for global re-localization, mitigating the problems of perceptual aliasing and improving localization accuracy.

Localization trajectories and 6-DoF pose estimation of a sequence with 100 length are given in Figure 9. It further shows that the localization result is smooth and accurate. The corresponding estimation of the 6-DoF poses on x, y, z, roll, pitch and yaw is described in Figure 9b. It can be seen that the proposed method can track the ground truth accurately in terms of 6-DoF pose estimation. This is of importance when using the localization result for re-localization and loop closure detection.

Figure 10 illustrates the distribution and histogram of the re-localization errors (mean squared errors) of all sequences with 100 length. Statistically more than half of poses estimated by the proposed method are within 20 meters, while this is less than $15\%$ percentage for Posenet. Moreover, there are some big errors, e.g., more than 200 meters, of Posenet, which indicates that it may have perceptual aliasing problems during pose estimation. It tends to be common in this challenging dataset, as shown in Figure 7. Therefore, it is verified that the recurrent model encapsulating the relationship between consecutive image frames is effective for global re-localisation using a video clip.

## 4. Conclusion

We have presented an approach for 6-DoF video-clip re-localization that exploits the temporal dependencies in the video stream to improve the localization accuracy of the



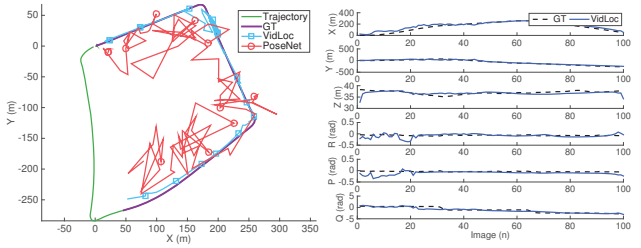

(a) Sequence length: 10.

(b) Sequence length: 20.

(c) Sequence length: 50.

(d) Sequence length: 100.

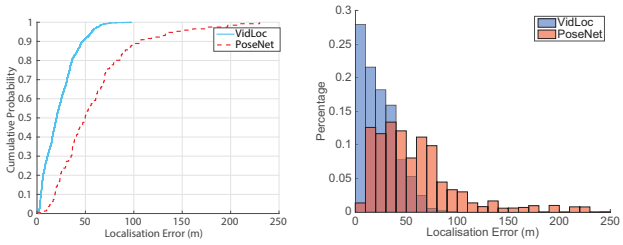
Figure 8: Global localization results on different lengths of sequences superimposed on Google Map.




(a) Localization results.

(b) Estimates of 6-DoF poses.

Figure 9: Localization result and estimates of 6-DoF poses of a sequence with 100 length.




(a) Distribution of errors.

(b) Histogram of errors.

Figure 10: Distribution and histogram of localization errors of all sequences with 100 length.

global pose estimates. We have studied the impact of window size and shown that our method outperforms the closest related approaches for monocular RGB localization by a fair margin.

For future work we intend to investigate means of making better use of the depth information, perhaps by forcing the network to learn to make use of geometrical information. One means of doing this would be to try predict the scene coordinates of the input RGB-D image using the CNN in an intermediate layer and then derive the pose from this and the input image. In essence this would be like unifying appearance-based localization and geometry-based localization in one model.

## References

[1] S. Bianco, M. Buzzelli, D. Mazzini, and R. Schettini. Logo recognition using cnn features. In *International Conference on Image Analysis and Processing*, pages 438–448. Springer, 2015.

[2] C. M. Bishop. Mixture density networks. 1994.

[3] E. Brachmann, F. Michel, A. Krull, M. Ying Yang, S. Gumhold, et al. Uncertainty-driven 6d pose estimation of objects and scenes from a single rgb image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3364–3372, 2016.

[4] R. Clark, S. Wang, H. Wen, A. Markham, and N. Trigoni. Vinet: Visual-inertial odometry as a sequence-to-sequence learning problem. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[5] Y. Gal and Z. Ghahramani. Bayesian convolutional neural networks with bernoulli approximate variational inference. *arXiv preprint arXiv:1506.02158*, 2015.

[6] T. Goldstein, P. Hand, C. Lee, V. Voroninski, and S. Soatto. Shapefit and shapekick for robust, scalable structure from motion. In *European Conference on Computer Vision*, pages 289–304. Springer, 2016.

[7] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund. Particle filters for positioning, navigation, and tracking. *IEEE Transactions on signal processing*, 50(2):425–437, 2002.

[8] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[9] A. Kendall and R. Cipolla. Modelling uncertainty in deep learning for camera relocalization. *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2016.

[10] A. Kendall, M. Grimes, and R. Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2938–2946, 2015.

[11] T. Kroeger and L. Van Gool. Video registration to sfm models. In *European Conference on Computer Vision*, pages 1–16. Springer, 2014.

[12] W. Maddern, G. Pascoe, C. Linegar, and P. Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *The International Journal of Robotics Research (IJRR)*, to appear.

[13] S. Middelberg, T. Sattler, O. Untzelmann, and L. Kobbelt. Scalable 6-dof localization on mobile devices. In *European Conference on Computer Vision*, pages 268–283. Springer, 2014.

[14] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pages 127–136. IEEE, 2011.

[15] P. Newson and J. Krumm. Hidden markov map matching through noise and sparseness. In *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*, pages 336–343. ACM, 2009.

[16] T. Sattler, B. Leibe, and L. Kobbelt. Fast image-based localization using direct 2d-to-3d matching. In *2011 International Conference on Computer Vision*, pages 667–674. IEEE, 2011.

[17] T. Sattler, B. Leibe, and L. Kobbelt. Improving image-based localization by active correspondence search. In *European Conference on Computer Vision*, pages 752–765. Springer, 2012.

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

[19] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 806–813, 2014.

[20] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2930–2937, 2013.

[21] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[22] N. Sunderhauf, S. Shirazi, A. Jacobson, F. Dayoub, E. Pepperell, B. Upcroft, and M. Milford. Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free. *Proceedings of Robotics: Science and Systems XII*, 2015.

[23] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[24] W. Zhang and J. Kosecka. Image based localization in urban environments. In *3D Data Processing, Visualization, and Transmission, Third International Symposium on*, pages 33–40. IEEE, 2006.

[25] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*, pages 487–495, 2014.