End-to-End, Sequence-to-Sequence Probabilistic Visual Odometry through Deep Neural Networks

Citation for published version:

Digital Object Identifier (DOI):
10.1177/0278364917734298

Link:
Link to publication record in Heriot-Watt Research Portal

Document Version:
Peer reviewed version

Published In:
International Journal of Robotics Research

General rights
Copyright for the publications made accessible via Heriot-Watt Research Portal is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
Heriot-Watt University has made every reasonable effort to ensure that the content in Heriot-Watt Research Portal complies with UK legislation. If you believe that the public display of this file breaches copyright please contact open.access@hw.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.
Abstract

This paper studies Visual Odometry (VO) from the perspective of Deep Learning. After tremendous efforts in the robotics and computer vision communities over the past few decades, state-of-the-art VO algorithms have demonstrated incredible performance. However, since the VO problem is typically formulated as a pure geometric problem, one of still missing key features of current VO systems is the capability to automatically gain knowledge and improve performance through learning. In this paper, we investigate whether Deep Neural Networks can be effective and beneficial to the VO problem. An End-to-end, Sequence-to-sequence Probabilistic Visual Odometry (ESP-VO) framework is proposed for the monocular VO based on deep Recurrent Convolutional Neural Networks (RCNNs). It is trained and deployed in an end-to-end manner, i.e., directly inferring poses and uncertainties from a sequence of raw images (video) without adopting any modules from the conventional VO pipeline. It can not only automatically learn effective feature representation encapsulating geometric information through Convolutional Neural Networks, but also implicitly model sequential dynamics and relation for VO using deep Recurrent Neural Networks. Uncertainty is also derived along with the VO estimation without introducing much extra computation. Extensive experiments on several datasets representing driving, flying and walking scenarios show competitive performance of the proposed ESP-VO to the state-of-the-art methods, demonstrating promising potential of Deep Learning technique for VO and verifying that it can be a viable complement to current VO systems.

Keywords

Visual odometry, pose estimation, uncertainty, deep learning, recurrent convolutional neural networks

1 Introduction

Visual odometry (VO) has attracted significant interest in both the robotics and computer vision communities over the past few decades (Scaramuzza and Fraundorfer 2011). As one of fundamental elements of many tasks in robotics and computer vision, VO has been widely applied to various applications, ranging from self-driving cars and autonomous drones to virtual and augmented reality. In general, the VO methods can be divided into two types in terms of the camera used: stereo VO and monocular VO. Since a single camera is cheaper, lighter and more widespread than a stereo rig and the stereo VO degenerates to the monocular one when the ratio of stereo baseline to depth is extremely small, this work focuses on monocular VO.

Enormous work has been done to develop an accurate, robust and reliable monocular VO system. As shown in Figure 1(a), a classic pipeline (Scaramuzza and Fraundorfer 2011; Fraundorfer and Scaramuzza 2012), which typically consists of camera calibration, feature detection, feature matching (or tracking), outlier rejection (e.g., RANSAC), motion estimation, scale estimation and optimisation (e.g., Bundle Adjustment (BA)), has been developed and broadly recognised as a standard VO framework to follow. Some state-of-the-art algorithms based on this pipeline have demonstrated excellent performance in terms of accuracy (Song et al. 2013; Mur-Artal et al. 2015). However, one of appealing yet still missing features of current VO systems is the capability to automatically gain knowledge to improve performance in terms of accuracy and robustness during continuous tests and/or daily usage. The current VO systems heavily rely on manual troubleshooting to analyse
failure cases, trace faults and refine localisation results. In consequence, there could be some corner cases still left unconsidered. Furthermore, it is difficult for current VO systems to benefit from cases where training data (i.e., accurate trajectories along with the perceived images) is available. This data is often easily collectable in certain situations, for example, when LiDAR or stereo VO setup is available. Currently, this data is mainly used in the debugging procedure and is mostly discarded when the VO system is in operation. To tackle this problem, there have been some attempts to learn VO by using Machine Learning (ML) algorithms, such as Gaussian Processes (Guizilini and Ramos 2013) and Support Vector Machines (Ciarfuglia et al. 2014). Since the traditional ML techniques tend to have limited ability to directly learn from extensive, high-dimensional data (e.g., raw images) (LeCun et al. 2015), the previous algorithms of VO all use some means of feature extraction (like optical flow) rather than raw images, which limits the ability to harness the rich information contained in the raw data. Besides, it is also challenging for these methods to truly exploit large-scale image data. These suggest that a more powerful ML method would be beneficial to the VO problem.

Taking advantage of an overwhelming availability of data, Deep Learning (DL) (Goodfellow et al. 2016) has recently been dominating many computer vision tasks, such as object recognition and detection, and applied to robotic applications, e.g., robotic grasping and manipulation (Lenz et al. 2015), with promising or even superhuman performance. However, it has not dominated VO or pose estimation yet. In fact, DL is mostly employed in 2D vision related problems and there are limited works on VO and 3D geometry. We presume that there are two main reasons for this. First, most of the existing DL architectures and pre-trained models are essentially designed to tackle recognition and classification problems, which drives deep Convolutional Neural Networks (CNNs) to extract high-level appearance features from images. In the context of VO, learning visual appearance eventually confines it to function only in trained environments and seriously hinders its generalisation. This is a reason why most of standard VO algorithms heavily rely on geometric features rather than appearance ones to enable them to work in different scenarios. Therefore, for effective VO, Deep Neural Networks (DNNs) need to learn new and efficient feature representation encapsulating geometric information.

Moreover, a VO method ideally should model motion and dynamics by examining the changes and connections among consecutive images instead of processing a single image, which is very different from object recognition and classification, such as ImageNet challenge. This is because a robot usually works continuously, producing sequential imagery over time for the VO to estimate poses. In a sequence of images, there is much information to interpret and exploit, and perhaps it is more than what current methods are using and modelling. Therefore, it is necessary for the DNNs to perform sequential learning for VO, which the CNNs alone are inadequate for.

In this paper, a novel DL based monocular VO framework, termed End-to-end, Sequence-to-sequence Probabilistic Visual Odometry (ESP-VO), is proposed by leveraging deep Recurrent Convolutional Neural Networks (RCNNs) (Donahue et al. 2016). Our main contributions are as follows:

- We show that it is feasible to achieve monocular VO in an end-to-end manner based on DL, i.e., directly estimating poses from raw images. To the best of our knowledge, this is the first end-to-end approach on the monocular VO based on DNNs.
- A RCNN architecture is developed to generalise the VO algorithm to totally new environments by using feature representation learned by the CNNs.
- Sequential dependence and complex motion dynamics of an image sequence, which are critical to the VO but difficult to be explicitly or easily modelled by human, are implicitly encapsulated and automatically learned by deep Recurrent Neural Networks (RNNs) to perform sequence-to-sequence pose estimation.
- Uncertainty of the VO is derived along with pose estimation in an unsupervised form without introducing much extra computation.

Because the proposed ESP-VO is realised in an end-to-end fashion, it does not require common modules in the standard VO pipeline, e.g., camera calibration, sparse feature extraction and matching. It also has some appealing properties, such as working reliably in low-texture environments even with rolling-shutter cameras and recovering accurate scales for monocular VO. This paper extends the work presented in (Wang et al. 2017) with significantly more details on the architecture and training of the DNN, new contribution on uncertainty estimation, and extensively more evaluations on mobile robots and extended to flying robots and human motion.

The rest of this paper is organised as follows. Section 2 reviews related work on VO. The monocular VO algorithm and uncertainty estimation are described in Section 3 and Section 4, respectively. Section 5 investigates cost function and optimisation. Experimental results are given in Section 6, followed by discussions on the DL based VO in Section 7. Finally, conclusions are drawn in Section 8.

2 Related work

Monocular VO and its close counterparts, visual Simultaneous Localisation and Mapping (SLAM) and Structure from Motion (SfM), have been extensively investigated in both
computer vision and robotics. Existing work on them is reviewed in this section.

There are mainly two types of VO algorithms in terms of the technique and framework adopted: geometry based methods and learning based methods (Ciarfuglia et al. 2014). The geometric methods, which are built up on multiple-view geometry (Hartley and Zisserman 2003) and photometric consistency, have a long history and still dominate the area of VO, while learning based methods based on ML techniques are recently emerging as data-driven approaches and become increasingly popular. They both have their own pros and cons.

2.1 Methods based on geometry

Geometry based methods relying on geometric constraints extracted from imagery to estimate motion have been well established based on rigorous principles, and have achieved tremendous success. Hence, most of state-of-the-art VO algorithms fall into this family. They can be further divided into sparse feature based methods and dense/direct methods according to whether images are pre-processed with sparse feature extraction and matching.

2.1.1 Sparse feature based methods

Sparse feature based methods solve the VO problem in two main steps, as shown in Figure 1(a). First, salient feature points are extracted from the raw images and matched to find feature point correspondences among images. Then, the poses are estimated based on the
geometric relationship of the feature points. To mitigate
the effects from noise and outliers, outlier rejection, such
as RANSAC, and local optimisation are necessary. Nistér
et al. (2004) propose one of the first VO systems by
estimating VO based on a stereo rig. Stereo matching
is applied for robust VO algorithm in (Geiger et al.
2011). However, all VO algorithms suffer from drifts
over time. To overcome this problem, visual SLAM/SIM
can be adopted to maintain a feature map for drift
correction along with pose estimation. Davison et al. (2007)
propose MonoSLAM, the first real-time monocularity visual
SLAM, by estimating sparse features as system states
in the framework of Extended Kalman Filter (EKF). On
the other hand, Parallel Tracking and Mapping (PTAM),
a keyframe based monocular SLAM, separates camera
tracking and feature mapping into different procedures
(Klein and Murray 2007). Both the filtering based and
keyframe based methods achieve excellent performance
in small workspaces. However, when transferring these
approaches to large-scale environments, there are more
challenging problems, e.g., the well-known scale drifts
of motion estimation and map in monocular vision (Strasdat
2012). To alleviate this, a scale-drift aware method based
on similarity transformation is proposed in (Strasdat et al.
2010) by introducing scale factors into keyframe based
optimisation and correcting the drifts upon loop closure.
Based on these, Mur-Artal et al. (2015) recently propose the
ORB-SLAM, one of the state-of-the-art monocularity visual
SLAM algorithms based on sparse features, demonstrating
excellent performance in various scenarios. Since the
trajectory and map are reconstructed up to scale in the
monocular VO and visual SLAM, some extra information
or device is required to recover an absolute scale. For
example, (Song et al. 2013, 2016) utilises the fixed height
of the camera to the ground to estimate the scale. For
more details on sparse feature based methods, we refer the
reader to (Scaramuzza and Fraundorfer 2011; Fraundorfer
and Scaramuzza 2012).

The feature based methods are sensitive to the outliers of
feature matching, which can be caused by image motion
blur, appearance similarity, occlusion, etc. The other big
limitation is that they only use salient features without
benefiting from rich information in the whole image. This
is a reason why dense/direct methods have been proposed.

2.1.2 Dense/Direct methods

Directly optimising photometric errors across the whole
images under the assumption of photometric consistency,
dense/direct methods do not employ manually designed
feature point detection, descriptor or matching, see Figure
1(b). Dense Tracking and Mapping (DTAM) first demon-
strates real-time dense monocular SLAM (Newcombe et al.
2011). Semi-dense/direct approaches which realise superior
performance in large-scale environments are also devel-
oped for the monocularity VO in (Engel et al. 2013; Forster
et al. 2014). Large-Scale Direct Monocular SLAM (LSD-
SLAM), one of the state-of-the-art monocular direct visual
SLAM methods, uses keyframe based photometric align-
ment and pose graph SLAM to estimate poses and build
consistent point cloud maps (Engel et al. 2014). To handle
the scale drift problem, it extends the scale-drift aware
method in (Strasdat et al. 2010) to an image alignment
one based on different keyframes. In recent work, a novel
direct sparse VO paradigm is designed to not only minimise
the photometric errors, but also estimate model parameters
by sampling pixels across the images (Engel et al. 2016).
Note that in our work the dense and direct concepts are
not distinguished although Engel et al. (2016) discuss the
difference between them in detail.

Since the dense/direct methods tend to be more accurate
in principle than sparse feature based ones and can work
better in texture-less environments by exploiting all the
pixels in the images, they are increasingly gaining more
favour. However, their main drawback is that the accuracy
of the photometric alignment is severely degraded when
the baseline of two matching images is long, due to the
increased number of local minima under long baseline
(Engel et al. 2013). This is common when robot moves fast
and camera frame rate is relatively low. We will discuss this
more in Section 6.

2.2 Methods based on learning

Learning based methods aim to derive motion model
and infer VO from sensor data by using ML techniques
without explicitly applying geometric theory. As data-
driven approaches, most of them are based on supervised
learning, and a large amount of labelled data (label for the
VO could be ground truth poses) is often demanded to train
proper models for VO.

2.2.1 Traditional learning based methods

Some work based on traditional ML techniques has been
conducted to solve the monocular VO problem. Sparse
optical flow is used to train K Nearest Neighbour (KNN),
Gaussian Processes (GP) and Support Vector Machines
(SVM) regression algorithms for monocular VO in (Roberts
et al. 2008), (Guizilini and Ramos 2013) and (Ciarfuglia
et al. 2014), respectively. Specifically, in (Roberts et al.
2008), the optical flow of the whole image is trained
separately in several small grids by using KNN to reduce
the dimensionality of the model. Then, the motion is
predicted by voting these optical flow estimates. It shows
that the learning based method can potentially estimate
the VO without camera calibration. Guizilini and Ramos
(2013) propose a multiple-output GP method which can
infer linear and angular velocities after training with optical
flow. Thanks to the GP framework, it can also estimate the covariance matrices of the translation and rotation. SVM is also applied to train regression models for the monocular VO by using optical flow in (Ciarfuglia et al. 2014).

These methods demonstrate an interesting direction and some promising results of achieving the monocular VO by learning instead of manual design. However, since the learning-based methods are recently emerging, there is limited amount of work and no one has directly dealt with raw images yet. This could be because the traditional ML techniques are widely recognised inefficient when encountering extensive high-dimensional data (LeCun et al. 2015), e.g., a large number of images. In order to tackle this problem, new powerful ML methods are necessary.

2.2.2 Deep Learning based methods

DL, which can make full use of large-scale dataset of images to automatically learn suitable feature representation for a task, has been increasingly applied to various computer vision and robotics problems, showing remarkable results.

CNNs Recently, CNNs have increasingly been used to tackle geometric problems, e.g., predicting depth (Eigen and Fergus 2015)(Liu et al. 2016)(Garg et al. 2016) and surface normal (Bansal et al. 2016) from a single image. In order to estimate geometric information related to motion, a pair of images is usually provided for CNNs. Ilg et al. (2017) propose a stacked CNN architecture for accurate optical flow estimation by using two images. Multiple stacked encoder-decoder convolutional networks are also designed to estimate depth and motion concurrently in (Ummenhofer et al. 2017).

DL has achieved impressive results on robot localisation. Features of CNNs, for instance, have been utilised for appearance-based place recognition in challenging environments (Sünderhauf et al. 2015b; Shahid et al. 2016) and semantic mapping (McCormac et al. 2016). However, there has been surprisingly little work on the VO. To our knowledge, DL based VO is first realised in (Konda and Memisevic 2015) for stereo vision through synchrony detection between image sequences and features. After estimating depth from stereo images, the CNN predicts discretised changes of direction and velocity by using softmax function. Although this work provides a feasible solution to the DL based stereo VO, it inherently formulates the stereo VO as a classification problem rather than pose regression. In (Agrawal et al. 2015), the VO problem is also considered as a classification problem by using features learned from ego-motion. Camera relocalisation using a single image is studied in (Kendall et al. 2015b) by fine-tuning pre-trained models of ImageNet CNNs with images of a specific scene. A trained neural network, PoseNet, serves as a “map” whose size does not increase proportionally to the size of the environment, which is a great advantage compared with point cloud maps or image database. PoseNet is further improved in (Jia et al. 2016) and (Kendall and Cipolla 2017) by training with different pre-trained models and geometric loss functions, respectively. However, because a trained CNN model serves as an appearance “map” of the scene, the network needs to be trained from scratch or at least fine-tuned for a new environment, which restricts the technique to work only in limited areas. This is also one of the biggest difficulties when applying DL for VO. To overcome this problem, the CNNs are provided with dense optical flow instead of raw images for motion estimation in (Costante et al. 2016). Three different architectures of CNNs are developed to learn appropriate features for the VO, achieving robust performance even with blurred and under-exposed images. However, the proposed CNNs require pre-processed dense optical flow as input, which cannot benefit from the end-to-end learning on raw images.

Muller (2016) also proposes a CNN-based model by using optical flow images produced from a DNN (Dosovitskiy et al. 2015). In order to explicitly incorporate geometric transformation into DNNs, some geometric computations are implemented as network layers in (Handa et al. 2016), which demonstrates some promising preliminary results on RGB-D VO and dense image alignment using a siamese network. Very recently, Zamir et al. (2016) propose a CNN framework to learn generic 3D representation by performing camera pose estimation and matching between pairs of image patches, achieving state-of-the-art results on wide baseline feature matching. Since it focuses on how to estimate a pose by using a pair of small image patches, it is different from the VO problem which predicts poses in a long trajectory through a sequence of images. Similarly, a pair of full images is used in (Melekhov et al. 2017) for wide-baseline relative pose estimation based on a hybrid CNN architecture.

Because CNNs are incapable of sequential learning, none of the previous work considers using image sequences or videos for the VO or pose estimation. In this work, we tackle this by leveraging the powerful combination of CNNs and deep RNNs.

RNNs Sequential learning and modelling are critical to the applications involving time-series data. This has been extensively proved by extraordinary successes of applying RNNs to speech recognition (Graves and Jaitly 2014), language translation (Sutskever et al. 2014), video description (Donahue et al. 2016), multi-target tracking (Ondruska and Posner 2016; Milan et al. 2017), video relocalisation (Clark et al. 2017a), visual inertial odometry (Clark et al. 2017b), etc. For VO, sequential dependence, dynamics and motion implied in a sequence of images are also of importance, and considering only a single image greatly hampers the ability to achieve accurate
pose estimates. Therefore, in the subsequent sections, a sequence-to-sequence monocular VO will be proposed to estimate a sequence of poses and uncertainties directly from a sequence of raw images instead of being restricted to frame-to-frame only.

**Uncertainty Estimation in DNNs** Uncertainty plays an important role in robotics, especially for localisation and navigation (Thrun et al. 2005). In geometry based VO and visual SLAM systems, the uncertainty or covariance can not only be used to fuse with other sensor modalities but also considerably facilitate feature tracking and active search.

Determining uncertainty from DNNs in an efficient manner remains an open and active research area. Gal and Ghahramani (2016) attempt to obtain uncertainty from the DNNs by multiple predictions with dropout. Rigours mathematical proof is given to prove that it is a variational approximation to Gaussian process. The technique has also been applied to improve camera relocalisation (Kendall and Cipolla 2016) and image segmentation (Kendall et al. 2015a). However, as a sampling approach inherently, it requires a large number of executions to have enough samples for unbiased uncertainty estimation, especially when recovering covariance on 6 DoF VO estimation. Therefore, it may not be applicable for real-time robot localisation and VO. Moreover, it is not necessary that every DNN uses dropout although it is widely used to prevent overfitting.

The previous work based on dropout represents uncertainty in a non-parametric form, while the other approach is parametric, such as covariance in Kalman filtering. This explicit representation of uncertainty can be more efficient in some cases. Bishop (1994) proposes Mixture Density Networks (MDNs) as a solution to predict parameters of a distribution rather than sole mean prediction. Therefore, it is possible to estimate uncertainty of DNNs based on the MDNs. A recent great work (Kendall and Gal 2017), which is available after the submission of this paper, gives good insights and comprehensive discussion on uncertainty in Deep Learning models. According to its taxonomy, the uncertainty proposed in our work belongs to heteroscedastic aleatoric uncertainty, which captures noise inherent in system observations.

### 3 Monocular Visual Odometry through RCNN

In this section, the deep RCNN framework realising the monocular VO in an end-to-end, sequence-to-sequence fashion is described in detail. It is mainly composed of CNN based feature extraction, RNN based sequential learning, fully-connected (FC) layer and SE(3) composition layer.

#### 3.1 Network architecture

There have been some popular and powerful DNN architectures, such as VGGNet (Simonyan and Zisserman 2014) and GoogLeNet (Szegedy et al. 2015), developed for computer vision tasks, producing remarkable performance. Most of them are designed with tackling recognition, classification and detection problems in mind, which means that they are trained to learn knowledge from appearance and image context. However, as discussed before, VO which is rooted in geometry should not be closely coupled with appearance. Therefore, it is difficult to simply adopt current popular DNN models for the VO problem. A framework which can learn geometric feature representations is of importance to address the VO and other geometric problems. Meanwhile, it is essential to derive connections among consecutive image frames, e.g., motion models, since VO systems evolve over time and operate on image sequences acquired during movement. Therefore, the proposed RCNN takes these two requirements into consideration.

The architecture of the proposed end-to-end VO system is shown in Figure 1(c). It takes a video clip or a sequence of monocular images as input. At each time step, the image frame is pre-processed by subtracting the mean RGB channel values of the training set and, optionally, resizing to a new size in the multiple of 64. Two consecutive images are stacked together to form a tensor for the deep RCNN to learn how to extract motion information and estimate poses. Specifically, the image tensor is fed into a CNN to produce an effective feature, which is then passed through a RNN for sequential learning. FC layers are introduced to transform RNN output to odometry estimates with uncertainty, followed by an SE(3) composition layer that performs pose composition on Special Euclidean Group. The network is trained with ground truth poses of training images to predict their poses by using features and models learned. Each image pair yields a 6 Degree-of-Freedom (DoF) pose estimate and its covariance at each time step through the network. Note that only poses have training labels (ground truth), while uncertainties are derived without supervision. Details on how to estimate uncertainties will be given in Section 4. Based on this architecture, the VO system develops over time and estimates poses as new images are captured.

The advantage of the proposed RCNN based architecture is to allow simultaneous feature extraction and sequential modelling of VO through the combination of powerful capabilities of both CNNs and RNNs. Note that no camera calibration or extra information is needed for the proposed VO system, which differs from geometry based methods.
3.2 Feature extraction: CNN

In order to automatically learn effective features that are suitable for the VO problem, a CNN is developed to perform feature extraction on the concatenation of two consecutive monocular images. The feature representation is ideally geometric instead of being associated with appearance or visual context because VO systems need to be generalised to and deployed in unknown environments. The structure of the CNN is inspired by the network for optical flow estimation in (Dosovitskiy et al. 2015).

The configuration of the CNN is outlined in Table 1 and an example of the dimensions of its tensors on KITTI dataset is given in Figure 2. It takes a tensor generated by stacking two consecutive images as input. It is composed of 9 convolutional layers and each layer is followed by a Rectified Linear Unit (ReLU) non-linearity activation except Conv6, i.e., 17 layers in total. The sizes of the receptive fields in the network gradually reduce from $7 \times 7$ to $5 \times 5$ and then $3 \times 3$ to capture small interesting features. Zero-paddings are introduced to either adapt to the configurations of the receptive fields or preserve the spatial dimension of the tensor after convolution. The number of the channels, i.e., the number of filters for feature detection, increases to learn various features.

![Fig. 2. Architecture of input and CNN of proposed monocular VO system. The dimensions of the tensors shown here are given as an example based on the image size of the KITTI dataset. The CNN ones should vary according to the size of the input image. Camera image credit: KITTI dataset.](image)

![Fig. 3. Architecture of RNN, fully-connected (FC) layer and SE(3) composition layer of proposed monocular VO system (continued from Figure 2).](image)

<table>
<thead>
<tr>
<th>Layer</th>
<th>Receptive Field Size</th>
<th>Padding</th>
<th>Stride</th>
<th>Number of Channels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conv1</td>
<td>$7 \times 7$</td>
<td>3</td>
<td>2</td>
<td>64</td>
</tr>
<tr>
<td>Conv2</td>
<td>$5 \times 5$</td>
<td>2</td>
<td>2</td>
<td>128</td>
</tr>
<tr>
<td>Conv3</td>
<td>$5 \times 5$</td>
<td>2</td>
<td>2</td>
<td>256</td>
</tr>
<tr>
<td>Conv3_1</td>
<td>$3 \times 3$</td>
<td>1</td>
<td>1</td>
<td>256</td>
</tr>
<tr>
<td>Conv4</td>
<td>$3 \times 3$</td>
<td>1</td>
<td>2</td>
<td>512</td>
</tr>
<tr>
<td>Conv4_1</td>
<td>$3 \times 3$</td>
<td>1</td>
<td>1</td>
<td>512</td>
</tr>
<tr>
<td>Conv5</td>
<td>$3 \times 3$</td>
<td>1</td>
<td>2</td>
<td>512</td>
</tr>
<tr>
<td>Conv5_1</td>
<td>$3 \times 3$</td>
<td>1</td>
<td>1</td>
<td>512</td>
</tr>
<tr>
<td>Conv6</td>
<td>$3 \times 3$</td>
<td>1</td>
<td>2</td>
<td>1024</td>
</tr>
</tbody>
</table>
The CNN takes raw images instead of pre-processed counterparts, such as optical flow or depth images, as input because the network is trained to learn an efficient feature representation with reduced dimensionality for VO. This learned feature representation not only compresses the original high-dimensional image into a compact description, but also boosts the successive sequential training procedure. Hence, the last convolutional feature Conv6 is passed to the RNN for sequential modelling.

### 3.3 Sequential learning: Deep RNN

In the last few years, there has been incredible success on applying RNNs to tackle different problems, such as image/video captioning and language modelling (speech recognition, machine translation, etc.). Since RNNs are capable of passing previous information for current state estimate and of modelling dependencies in a sequence, it is well suited to the VO problem which involves both temporal model (motion model) and sequential data (image sequence). Estimating pose of current image frame, for instance, can benefit from information encapsulated in previous frames and states. In fact, this insight has already existed in the conventional VO systems: multi-view geometry is able to avoid some issues of two-view geometry (Hartley and Zisserman 2003). However, RNN is not suitable to directly learn a sequential representation from high-dimensional raw images. Therefore, the RNN of the proposed system adopts the CNN features instead.

Following the CNN, a deep RNN is designed to conduct sequential learning, i.e., to model dynamics and connections by using a sequence of CNN features. This modelling is performed implicitly by the RNN to automatically discover appropriate sequential knowledge. Therefore, it may generate models that are not limited to physical movement and geometry. RNNs are different from CNNs in that they maintain memory of their hidden states over time and have feedback loops among them, which enables their current hidden state to be related to the previous ones. Hence, RNNs can find out the connections among the current input and the previous inputs and states in the sequence. Given a convolutional feature \( I_k \) at time \( k \), a RNN updates at time step \( k \) by:

\[
\begin{align*}
    h_k &= \mathcal{H}(W_{Ih}I_k + W_{hh}h_{k-1} + b_h) \\
    O_k &= W_{ho}h_k + b_O
\end{align*}
\]

where \( h_k \) and \( O_k \) are the hidden state and output at time \( k \) respectively, \( W \) terms denote corresponding weight matrices, \( b \) terms denote bias vectors, and \( \mathcal{H} \) is an element-wise non-linear activation function, such as sigmoid or hyperbolic tangent. Although in theory this standard RNN can learn sequences with arbitrary lengths, it is limited to short ones in practice due to the known vanishing gradient problem in back-propagation (Goodfellow et al. 2016).

In order to be able to explore and exploit correlations among images taken in long trajectories, Long Short-Term Memory (LSTM) which is capable of learning long-term dependencies by introducing memory gates and units (Hochreiter and Schmidhuber 1997; Zaremba and Sutskever 2014) is employed as our RNN, as shown in Figure 3. LSTM is able to explicitly determine which previous hidden states to be discarded or retained for updating the current state, being expected to learn motion and dynamics during pose estimation. The internal structure of a LSTM unit along with an unfolded LSTM over time is shown in Figure 4. It can be seen that the different memory gates control how information is obtained from previous states and passed to future. Specifically, given the convolutional feature \( I_k \) at time \( k \), the hidden state \( h_{k-1} \) and the memory cell \( c_{k-1} \) of the previous LSTM unit, the LSTM updates at time step \( k \) according to:

\[
\begin{align*}
    i_k &= \sigma(W_{Ii}I_k + W_{hi}h_{k-1} + b_i) \\
    f_k &= \sigma(W_{If}I_k + W_{hf}h_{k-1} + b_f) \\
    g_k &= \tanh(W_{Ig}I_k + W_{hg}h_{k-1} + b_g) \\
    c_k &= f_k \odot c_{k-1} + i_k \odot g_k \\
    o_k &= \sigma(W_{Io}I_k + W_{ho}h_{k-1} + b_o) \\
    h_k &= o_k \odot \tanh(c_k)
\end{align*}
\]

where \( \odot \) is element-wise product of two vectors, \( \sigma \) is sigmoid non-linearity, \( \tanh \) is hyperbolic tangent non-linearity, \( W \) terms denote corresponding weight matrices, \( b \) terms denote bias vectors, \( i_k, f_k, g_k, c_k \) and \( o_k \) are input gate, forget gate, input modulation gate, memory cell and output gate at time \( k \), respectively. Moreover, after unfolding the LSTM, each LSTM unit is associated with a specific time step, enabling the LSTM (RNN) to take input sequences of arbitrary lengths and produce corresponding pose estimates. This is appealing in practice since the VO system is not confined to fixed-length image sequences.
Although LSTM can handle long-term dependencies, it still needs depth on network layers to learn high level representation and model complex dynamics. The advantages of the deep RNN architecture have been proved in (Graves and Jaitly 2014) for speech recognition using acoustic signals. As shown in Figure 5, when a deep RNN with multiple layers is unfolded as a computational graph, it is not only deep along time, but also has deep structure on network layers by stacking several RNN layers with the hidden state $h$ of a RNN layer being input of the next one at each time step.

For the proposed network for VO, the deep RNN is constructed by stacking two LSTM layers as illustrated in Figure 3. Each of the LSTM layers has 1024 hidden states. With this deep LSTM component, the network is expected to learn temporal models and dynamics in the context of high-level CNN features. With the aid of this sequential learning, the features of the RNN encapsulates information on motion, which can be exploited for pose estimation.

### 3.4 Fully-connected layer

Two fully-connected layers are introduced to collect the features of the RNN and transfer them into motion and uncertainty. The first layer has 128 hidden states, followed by ReLU non-linearity activation. The other one contains 12 dimensions to represent both motion and uncertainty, i.e., 6 dimensions on motion $u$ with travelled distance $(\Delta x, \Delta y, \Delta z)$ and heading change $(\Delta \phi, \Delta \theta, \Delta \varphi)$ over last time step and 6 dimensions on their corresponding variances. With the aid of uncertainty estimates, this predicted motion can be used as odometry or dead-reckoning to fuse with other sensors, such as GPS and laser scanner, in the framework of Bayes filters. Next section will discuss how to obtain the uncertainty on the VO in detail. However, in order to derive the global poses with respect to the initial starting coordinate frame, the transformation on position and orientation needs to be composed over time. This is the reason why an SE(3) composition layer is designed.

#### 3.5 SE(3) composition layer

As a dead-reckoning technique, VO needs to incrementally compose odometry estimates in order to derive global poses of a camera with respect to the coordinate frame defined by an initial pose. A transformation $T$ in $\mathbb{R}^n$ can be represented as an element in the Special Euclidean Group $SE(n)$ as

$$ T = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} $$(3)

with rotation matrix $R \in SO(n)$ and translation $t \in \mathbb{R}^n$. The $SO(n)$ denotes Special Orthogonal Group. Then, concatenation of two 6 DoF poses can be conducted by matrix multiplication in the context of SE(3). For a sequence of poses, the current pose estimate is computed by composing the last one. Therefore, some network layers are required to perform this. Since RNNs have hidden states which can carry information from previous states, they should be capable of learning the pose composition. However, since there is no hyper-parameter which needs to be learned in this pose composition, a SE(3) composition layer is designed based on operations on SE(3) and directly introduced into the neural networks to accelerate learning. As shown in Figure 3, the previous pose estimate is explicitly connected to current time step. The key of developing this composition layer is to ensure that it can be processed though backpropagation for end-to-end training. We find this SE(3) composition layer is of importance to facilitate network convergence. Therefore, incorporating some geometric transformations as neural network layers can be helpful for VO, which is also reported in (Handa et al. 2016) encapsulating some geometric computations in computer vision as specific layers for neural networks.

### 4 Uncertainty estimation of VO

Uncertainty is usually derived in the framework of optimal state estimation (Thrun et al. 2005), especially based on Bayes filters. Assume that a robot system is described by a non-linear discrete-time process model as

$$ y_k = f(y_{k-1}, u_k, v_k) $$ (4)

where $u_k = [\Delta x_k, \Delta y_k, \Delta z_k, \Delta \phi_k, \Delta \theta_k, \Delta \varphi_k]^T$ is the motion input, $y_k$ is pose at time $k$, and $v_k \sim \mathcal{N}(0, Q_k)$. Figure 5. Deep Recurrent Neural Network. Top: 3-layer Deep RNN with recurrent connections. Bottom: Time-unfolded Deep RNN in which each unit is associated with a time step.
is a Gaussian process noise on motion input. In the context of VO, a camera is used as a sensor to calculate motion input $u_k$ for state prediction. If a pure VO system is discussed, there is no observation, and the drift and uncertainty of the VO system increase unboundedly over time (see experimental result in Figure 28). Therefore, other measurement sensors, such as GPS and laser scanner, or SLAM techniques are necessary to correct the drift and bound the uncertainty. Since camera can also be employed as a sensor for observation with other information, e.g., constant velocity model, as prediction, we clarify that the uncertainty of VO discussed in this work is the one on motion input, i.e., $v_k$, in order to avoid confusion with other vision based algorithms, such as visual SLAM and visual inertial systems. As aforementioned, this uncertainty is essential for a system using VO for sensor fusion (covariance propagation) or pose graph SLAM (covariance on odometry edge) and is more flexible than that of whole trajectory. If necessary, it can also be easily propagated to uncertainty on estimated poses or whole trajectory (Smith and Cheeseman 1986).

In order to obtain uncertainty for VO, we need to model feature extraction and camera geometry in a probabilistic perspective (Matthies and Shafer 1987; Davison 2005; Williams and Reid 2010; Strasdat 2012). It involves linearisation of camera projection function and coordinate transformation, Jacobian matrices calculations, etc., to transform the uncertainty on feature extraction to the one on VO estimation. This is non-trivial, especially maintaining an exact uncertainty estimation by taking all correlations into consideration. Therefore, it is common that this uncertainty is approximated by assuming that it is constant (Strasdat 2012; Mur-Artal et al. 2015).

Efficiently determining uncertainty from neural network is challenging and remains an open question. Firstly, most DNNs are designed and trained to only predict mean values. Secondly, uncertainty which is defined under probabilistic inference is not easily accessible as accurate labels for supervised training, e.g., no device directly measures the ground truth of uncertainty. Even if it can be computed based on Bayes filters, it is affected by many factors, such as linearisation points. This means that it may have to perform unsupervised learning on uncertainty estimation. Note that the uncertainty or probability of softmax function, which normalises probability among all classes for classification problems, is not sufficient for some applications (Gal and Ghahramani 2016). Intuitively, in the context of VO and robot localisation, the uncertainties involved are to model confidence on pose estimates, which are not necessarily to be classified into several classes.

### 4.1 VO uncertainty from DNNs

We show how to recover uncertainty of the VO from the DNNs in this section. Due to the dominating popularity of Gaussian distribution in robotics and computer vision, the uncertainty is modelled as multivariate Gaussian distribution although it can be easily extended to other distributions by using mixture models (Bishop 1994).

Regression DNNs is usually trained by minimising cost functions in the form of Mean Squared Error (MSE), producing only mean values given training data. This means that it is difficult to derive uncertainty, limiting its statistic information to inspect prediction. On the contrary, given a DNN and an image $x_k$ at time $k$ and assume that its prediction consists of both motion $u_k$ and its covariance $Q_k$ which together parametrise a conditional probability following Gaussian distribution, we have

$$p(u_k|x_k;\theta) = \frac{1}{\sqrt{(2\pi)^d |Q_k|}} \exp \left( -\frac{1}{2} (F_k(x_k;\theta) - u_k)^T Q_k^{-1} (F_k(x_k;\theta) - u_k) \right)$$

(5)

where $F_k(x_k;\theta)$ denotes the function defined by the neural network, which is parametrised by its parameters $\theta$ (weights and biases), to predict $u_k$ from $x_k$. If we maximise (5) by minimising the negative logarithm of it, it gives

$$\theta^* = \arg \max_{\theta} p(u_k|x_k;\theta)$$

$$= \arg \min_{\theta} \log |Q_k|$$

$$= \arg \min_{\theta} \log |Q_k| + (F_k(x_k;\theta) - u_k)^T Q_k^{-1} (F_k(x_k;\theta) - u_k)$$

(6)

We can see that MSE is a special case of this when $Q_k$ is assumed to be constant ($Q_k$ can be ignored since it does not affect the optimal value $\theta^*$) for all samples and time steps. Therefore, in order to have covariance of VO, $Q_k$ needs to be kept and predicted by the neural networks as the parameters of a Gaussian distribution. This is the reason why in the previous RCNN, the last fully-connected layer consists of both motion $u_k$ and its covariance $Q_k$, i.e., a subset of network outputs defines the VO estimate, while the remaining ones represent its uncertainty. Since most of current VO and visual SLAM methods assume that the covariance matrix $Q$ is diagonal with no correlation between motion input, e.g., constant velocity model in (Davison et al. 2007; Williams and Reid 2010; Strasdat et al. 2012; Mur-Artal et al. 2015), the covariance matrix is represented by $6$ standard deviations of motion vector as discussed in Section 3.4. Note that if necessary it is straightforward to estimate full covariance by predicting its correlation elements as extra states in the fully-connected layer.
Training without true covariances However, as aforementioned a big challenge is that in the framework of supervised learning, it is difficult to produce labels of \(Q\) for training although ones of \(u\) can be obtained by extra powerful devices, e.g., Vicon and laser scanner, or algorithms. In fact, according to (6), training a neural network by minimising the negative logarithm of (5) can be driven by minimising the differences between \(F(\mathbf{x}; \theta)\) and \(u\) without the need of labels of \(Q\). Since the value of covariance \(Q\) has to be also optimised to balance the cost function during this training procedure, it can be implicitly trained. This is different to MSE-based training approaches. Since no label is adopted to train the uncertainty estimation, it is unsupervised learning to some extent.

Maintaining the properties of covariance matrix Covariance matrix needs to be positive semi-definite. Hence, this property has to be maintained during optimisation. Some traditional ways, such as incorporating constraints into optimisation, are not suitable for training DNNs. Therefore, we propose to re-parametrise the covariance matrix into a lower-triangular matrix according to Cholesky decomposition, and then use the neural network to predict its elements instead. The covariance matrix can be recovered as a product of the predicted lower triangular matrix and its transpose. This not only ensures the positive semi-definite property, but also reduces the number of elements to be predicted from the neural network.

By designing the last fully-connected layer of the previous RCNN to incorporate states of both mean and covariance of a Gaussian distribution, the uncertainty of VO can be estimated along with motion prediction. Because this uncertainty estimation method is parametric and there is no much extra computation or layer to be introduced into the neural network, it tends to be more efficient compared to the dropout based sampling approach. Note that for uncertainty estimation there is no change on the architecture of the neural network except adding several additional units. However, the cost function for training the neural network needs to be re-formulated.

5 Cost function and optimisation

The DNN of ESP-VO is designed to simultaneously predict pose and uncertainty. Hence, as shown in Figure 6, the cost function for training is hybrid and composed of two parts, each of which deals with one type of prediction.

As for pose estimation, the proposed RCNN can be considered to compute the conditional probability of the poses \(\mathbf{Y}_t = (\mathbf{y}_1, \ldots, \mathbf{y}_t)\) given a sequence of monocular images \(\mathbf{X}_t = (\mathbf{x}_1, \ldots, \mathbf{x}_t)\) up to time \(t\):

\[
p(\mathbf{Y}_t|\mathbf{X}_t) = p(\mathbf{y}_1, \ldots, \mathbf{y}_t|\mathbf{x}_1, \ldots, \mathbf{x}_t) \tag{7}\]

Optimal network parameters \(\theta^*\) (weights and biases) for pose estimation can be learnt by maximising (7):

\[
\theta^* = \arg\max_{\theta} p(\mathbf{Y}_t|\mathbf{X}_t; \theta) \tag{8}
\]

Therefore, based on MSE, the Euclidean distance between the ground truth pose \(\mathbf{y}_k = (\mathbf{p}_k^T, \mathbf{\Phi}_k^T)^T\) and its estimate \(\hat{\mathbf{y}}_k = (\hat{\mathbf{p}}_k, \hat{\mathbf{\Phi}}_k)^T\) at time \(k\) can be minimised by

\[
\theta^* = \arg\min_{\theta} \frac{1}{t} \sum_{k=1}^{t} \|\mathbf{p}_k - \hat{\mathbf{p}}_k\|_2^2 + \kappa\|\mathbf{\Phi}_k - \hat{\mathbf{\Phi}}_k\|_2^2 \tag{9}
\]

where \(\mathbf{p}\) and \(\mathbf{\Phi}\) denote position and orientation respectively, \(\|\cdot\|\) is 2-norm, and \(\kappa\) is a scale factor to balance the weights of positions and orientations. \(\Phi\) is in quaternion representation in order to avoid problems of Euler angles in global coordinate frame.

In terms of the motion and uncertainty estimation, according to the discussion in Section 4.1 the RCNN is trained by maximising the conditional probability probability of the motions \(\mathbf{U}_t = (\mathbf{u}_1, \ldots, \mathbf{u}_t)\) given a sequence of monocular images \(\mathbf{X}_t = (\mathbf{x}_1, \ldots, \mathbf{x}_t)\) up to time \(t\):

\[
p(\mathbf{U}_t|\mathbf{X}_t) = p(\mathbf{u}_1, \ldots, \mathbf{u}_t|\mathbf{x}_1, \ldots, \mathbf{x}_t) \tag{10}\]

Hence, we have

\[
\theta^* = \arg\min_{\theta} \frac{1}{t} \sum_{k=1}^{t} \log \|\mathbf{Q}_k\| + (\mathbf{\hat{u}}_k - \mathbf{u}_k)^T \mathbf{Q}_k^{-1} (\mathbf{\hat{u}}_k - \mathbf{u}_k) \tag{11}
\]

where \(\mathbf{\hat{u}}\) and \(\mathbf{Q}\) are formulated from the last fully-connected layer. Since the SE(3) composition layer is a fixed computation without parameters, training label of \(\mathbf{u}_k\) can be obtained from \(\mathbf{y}_{k-1}\) and \(\mathbf{y}_k\) by reverse pose compounding. The orientation of \(\mathbf{u}\) is represented by Euler angles rather than quaternion since it tends to be small in local coordinate frame and does not have problems in global
frame. We also find that in practice using quaternion for this layer degrades the orientation estimate to some extent, which is also confirmed in the recent work (Zamir et al. 2016).

The cost function of the ESP-VO combines both (9) and (11), which jointly optimise the parameters of the neural network using backpropagation through time.

6 Experimental results

In this section, we evaluate the proposed VO method by comparing it with various state-of-the-art algorithms in different scenarios, ranging from outdoor driving car to indoor Micro Aerial Vehicle (MAV) and human motion.

6.1 Datasets

The datasets used for testing are collected with diverse hardware and platforms, representing different environments, movements, situations and applications. They cover outdoor and indoor, driving, flying and walking activities. Specifically, the datasets used are:

- Outdoor car driving scenario: KITTI VO dataset (Geiger et al. 2013), Málaga dataset (Blanco-Claraco et al. 2014) and Cityscapes dataset (Cordts et al. 2016)
- Indoor MAV scenario: EuRoC MAV dataset (Burri et al. 2016)
- Indoor motion scenario: Human motion dataset and NYU depth dataset (Silberman et al. 2012)

Sample images of the datasets are given in Figure 7

6.1.1 KITTI VO dataset

The KITTI VO/SLAM benchmark (Geiger et al. 2012, 2013), which was created during outdoor car driving activities, is one of the most well-known public datasets to evaluate VO and visual SLAM algorithms (Mur-Artal et al. 2015; Engel et al. 2015; Song et al. 2016). It has 22 sequences of images, of which 11 ones (Sequence 00-10) are associated with accurate ground truth. The other 10 sequences (Sequence 11-21) are only provided with raw sensor data without ground truth or GPS. Since this dataset was recorded at a relatively low image capture rate (10 frames per second) during driving in urban areas with many dynamic objects and the driving speed was up to 90 km/h, it is very challenging for monocular VO algorithms. Although the dataset contains imagery of four cameras, only the images of the left colour camera (Point Grey Flea 2 camera) are used in our experiments for monocular VO. The KITTI dataset represents applications on mobile robots, such as autonomous driving.

6.1.2 Málaga dataset

Málaga urban dataset (Blanco-Claraco et al. 2014), similar to the KITTI dataset, is collected in urban scenarios during driving. It provides 20 Hz stereo imagery along with data from LiDAR, GPS, etc. In this work, we only use the images of the left camera, and they are only employed to test pre-trained models without training. Since its image size is different from the KITTI’s, its images are cropped and resized to the KITTI image size.

6.1.3 Cityscapes dataset

Cityscapes dataset (Cordts et al. 2016) is designed for semantic urban scene understanding, e.g., dense semantic labelling and segmentation. It is also gathered when driving across cities. It contains a large number of stereo video clips captured in different cities for VO estimation. Similar to the Málaga dataset, the Cityscapes dataset is only used to test models.

6.1.4 EuRoC MAV dataset

The European Robotics Challenge (EuRoC) dataset (Burri et al. 2016) is a recently published dataset including visual inertial data collected by using a MAV. It has been increasingly used for visual inertial odometry, SLAM, 3D reconstruction, etc. Raw readings of an Inertial Measurement Unit (IMU), stereo images, ground truth and sensor calibration of 11 sequences are all provided for two environments, a machine hall and a Vicon room. It covers diverse motions of a flying robot from slow movements to
agile flights, producing high-quality images and motion-blurred, poor-exposed ones. More details of the dataset and configurations can be found in (Burri et al. 2016). Since the space of the Machine Hall is much bigger than the Vicon Room, we mainly use the 5 Machine Hall sequences. As in (Engel et al. 2015), for each sequence, the first about 200 images are excluded because they are taken for visual inertial calibration and some of their views are limited to floor. The EuRoC MAV dataset can be a good example of indoor applications using flying robots.

6.1.5 Self-collected indoor motion dataset

The indoor motion dataset was self-collected by a human walking in several indoor buildings. In order to train the neural networks, both images and their ground truth are necessary. The TUM RGB-D SLAM dataset (Sturm et al. 2012), as a high-quality dataset created in the scale of room, satisfies this well. However, it would be more interesting to perform the evaluation in large indoor environments instead of confined office rooms. Therefore, we collected images by using a hand-held Google Project Tango device, which is equipped with an IMU and a high frame-rate, fish-eye camera to provide pose estimates by using visual inertial localisation (Hesch et al. 2013). For the dataset, images are captured at 2 Hz as normal perspective rather than fish-eye wide-angle ones. Some sample images of the dataset are shown in Figure 7. Training data was mainly recorded in a departmental building with a big atrium, several common rooms and various corridors, while testing one was in another office building and a museum with a Café and walking people.

6.1.6 NYU depth dataset

NYU depth dataset V2 (Silberman et al. 2012), which is designed for indoor segmentation and scene understanding, is comprised of RGB-D video sequences captured by using a Microsoft Kinect in a variety of indoor scenes. Since the RGB camera of the Kinect is a rolling shutter one with narrow field of view and indoor spaces tend to have many areas with limited texture (e.g., walls and floors), the dataset is challenging for monocular VO and visual SLAM. The reactively fast motion of the camera makes it even more difficult for pure vision based VO. In order to evaluate our trained models in totally new environments, we employ this NYU dataset for testing. Note that there is no ground truth available for the sequences in this dataset.

6.2 Algorithms

Several state-of-the-art algorithms are compared in the subsequent experiments in order to evaluate the performance of the proposed VO method. Since visual SLAM becomes VO when no loop closure is detected, the algorithms to compare also include some visual SLAM algorithms with loop closure detection being disabled. Note that some results of the monocular VO/SLAM algorithms (ORB-SLAM and LSD-SLAM) are manually aligned against ground truth by similarity transformation since they are estimated up to an absolute scale.

6.2.1 VIS02

Open-source library VIS02 (Geiger et al. 2011) is one of the most popular VO algorithms, which is usually chosen as a baseline method for comparison and evaluation. It uses sparse feature based stereo matching to realise efficient monocular and stereo VO. Since monocular VO does not have an absolute scale, a fixed camera height (1.7 meters) is utilised by the monocular VIS02 to recover absolute positions. Therefore, the VIS02 is only employed in the experiments on the KITTI dataset. In contrast, its stereo version can directly obtain absolute poses and avoid scale drifts by using a fixed baseline of stereo vision. The stereo VO algorithm is also tested on the KITTI dataset for comparison.

6.2.2 Sparse feature based ORB-SLAM

As a recently developed state-of-the-art sparse feature based visual SLAM algorithm, ORB-SLAM (Mur-Artal et al. 2015) has demonstrated impressive results in different environments using monocular, stereo and RGB-D cameras. To achieve a comparison in the context of monocular VO, the ORB-SLAM algorithm is modified with its global loop-closure detection being disabled. Note that there is no other change and the loop closures of the co-visibility graph are still detected, followed by local BA. In the experiments, ORB-SLAM results are produced by setting the number of ORB features per image to 2000 and the fast threshold to 20. Constant velocity motion model is also used to produce more reliable results. Since the monocular ORB-SLAM algorithm does not recover an absolute scale, its keyframe trajectories are aligned to ground truth by using similarity transformation. The modification and alignment performed here are similar to these in (Engel et al. 2016). We stress that unless otherwise noted, the VO results of the ORB-SLAM in this work are generated from the modified VO version and do not represent the performance of the original SLAM version (although they should be the same when no loop exists).

6.2.3 Direct method based LSD-SLAM

LSD-SLAM (Engel et al. 2014) is a state-of-the-art monocular visual SLAM algorithm achieving superior performance based on the direct method. It not only detects scale drifts by conducting direct tracking, but also models noisy depth estimates in a probabilistic perspective to perform accurate tracking. Based on pose graph SLAM, LSD-SLAM can generate consistent maps of large-scale areas by using depth maps associated with keyframes. To provide results on monocular VO, similar to the ORB-SLAM, the LSD-SLAM algorithm is run by disabling its
global loop-closure detection. Similarity transformation of its results against ground truth is also applied to derive absolute poses. Note that in some experiments the VO estimates of the LSD-SLAM cannot be produced because the dataset is too challenging for direct method based tracking. More details will be discussed in the following experimental results.

6.3 Training and testing

Training DNNs requires a large amount of data to generalise well in reality and avoid overfitting. In this work, RCNNs take sequential images as input, which is different from popular CNNs based models using a single image. Unfortunately, sizes of current available datasets which can be used to train monocular VO are considerably smaller compared to the ones used for image recognition, etc. For example, the KITTI dataset has 21 sequences, while the famous ImageNet database contains over ten million images. Moreover, generating synthetic sequential data is more challenging when taking dynamics into consideration. In order to overcome this problem, we randomly select segments of the training image sequences with different lengths and starting and ending points, which potentially creates millions of different image sequences. This data augmentation technique is analogous to random image cropping usually used for training CNNs. Although the generated data is not as efficient as fresh ones for training the networks (because images, environments and motion exhibited in sequences are not new), we observe significant performance improvement on generalisation and overfitting by using this data augmentation method. The generated sequences are also randomly shuffled during training. Note that for image sequences in the context of VO, the shuffle is performed in terms of different image sequences instead of single images.

The network is implemented based on the TensorFlow framework and trained by using a NVIDIA Tesla K40/K80 GPU. Adam optimiser (Kingma and Ba 2014) is employed to train the network with starting learning rate 0.001 and parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$ (recommended values in (Kingma and Ba 2014)) for up to 200 epochs. Dropout, early stopping and incremental training techniques are introduced during training the networks. In order to reduce the number of training data needed and facilitate network convergence, the CNN is based on a pre-trained FlowNet model (Dosovitskiy et al. 2015). When a trained model is...

Fig. 8. Training losses and VO results of two models trained on KITTI dataset. Figures in the top and bottom rows are about the over-fitted and well-fitted models, respectively. (a) and (d): Training and validation losses. (b) and (e): Estimated VO on training data (Sequence 00). (c) and (f): Estimated VO on testing data (Sequence 05).
tested for VO estimation, a whole testing image sequence is fed into the network, producing a sequence of poses and covariances.

6.4 How overfitting affects VO results

It is well known that overfitting is an undesirable behaviour for Machine Learning based methods. However, its meaning and influence are unclear in the context of the VO problem. Concrete discussions on this, which can guide a better training on VO systems, are still missing. Some insights on our training procedure and results are described here. In Figure 8, the losses and VO results of two models (an over-fitted one and a well-fitted one) trained on the KITTI dataset are given. The big gap between the training and validation losses in Figure 8(a) indicates serious overfitting compared to the proper losses in Figure 8(d). Reflecting on the estimated VO of the training data, the results of the over-fitted model are much more accurate than those of the well-fitted model, as shown in Figure 8(b) and Figure 8(e). However, when the models are tested on testing data, the well-fitted model yields much better results, see Figure 8(c) and Figure 8(f). This is also very likely to happen when the model is deployed in practice working on other real data. Therefore, overfitting should be carefully examined when training a model for VO. Based on this example, it can be seen that for the DL based VO problem overfitting has very intuitive effects and can seriously degrade the odometry estimation. A well-fitted model is key to ensuring good generalisation and reliable pose estimation to untrained environments. During our work, it is found that orientation is more prone to overfitting than position. This could be because the orientation changes are usually smaller. In terms of underfitting, we assume this is rare because the capacity of a DNN is typically large and the size of training data tends to be limited.

Fig. 9. Average errors on translation and rotation against different path lengths and speeds. The ESP-VO model used is trained on Sequence 00, 01, 02, 08 and 09 and tested on Sequence 03, 04, 05, 06, 07 and 10. Its performance is expected to improve when it is trained on more data.
6.5 Results in outdoor driving scenario

There are two parts on outdoor car driving situation. First set of experiments is all based on KITTI dataset. We train and test models on KITTI dataset. Then, based on the model trained on KITTI dataset, we directly test on the Cityscapes and Málaga datasets without any further training or fine-tuning to evaluate the performance of our algorithm in new cities with different platforms.

6.5.1 Results on KITTI dataset

Two sets of experiments are conducted to evaluate the proposed method on the KITTI dataset. The first one is based on the Sequence 00-10 to quantitatively analyse its performance by using ground truth since the ground truth is only provided for these sequences. In order to have data for testing, only the Sequence 00, 01, 02, 08 and 09 which are relatively long are used for training. The trajectories are randomly segmented to different lengths to generate a large amount of data for training. The trained models are tested on the Sequence 03, 04, 05, 06, 07 and 10 for evaluation. Furthermore, because the ability to generalise well to new data is essential for DL based approaches, some experiments are conducted to analyse how the proposed method and trained VO models behave in totally new environments. For the VO problem, this is further required as aforementioned. Therefore, models trained on all the Sequence 00-10 are tested on new data, e.g., the Sequence 11-21 which do not have ground truth available and the raw data of KITTI benchmark. In order to have a quantitative comparison, the performance of the VO methods is analysed according to the KITTI VO/SLAM evaluation metrics, i.e., averaged Root Mean Square Errors (RMSEs) of the translational and rotational errors for all subsequences of lengths ranging from 100 to 800 meters and different speeds (the range of speeds varies in different sequences).

The ORB-SLAM, monocular VISO2 and stereo VISO2 are employed to evaluate the performance of the proposed VO method. Although we tried to compare against direct methods, LSD-SLAM consistently loses tracking for the KITTI dataset. This is because the images of the KITTI dataset are captured only at 10 Hz while driving at speeds of up to 90 km/h. Aligning images by minimising photometric
errors under a large baseline is very challenging for direct methods. Therefore, no result of LSD-SLAM is provided.

The first DL based model is trained on the Sequence 00, 01, 02, 08 and 09 and then tested on the Sequence 03, 04, 05, 06, 07 and 10. The average RMSEs of the estimated VO on the testing sequences are given in Figure 9 with the translation and rotation against different path lengths and speeds. Although the result of the ESP-VO is worse than that of the stereo VISO2, it is consistently better than the monocular VISO2 and ORB-SLAM except that the translational errors of the DL model on high speeds are slightly higher than the monocular VISO2. We presume that this is because the maximum velocities of the Sequence 00, 02, 08 and 09 are below 60 km/h and there is a very limited number of training samples whose speeds are bigger than 50 km/h (only some in Sequence 01). Without being trained with enough data covering the high-speed situation, the network tries to regress the VO but probably suffers from high drifts. It is interesting that the rotational errors become smaller on high velocities, which is opposite to

---

**Table 2. Results on Testing Sequences.** Only the best results of monocular VO are highlighted without considering stereo VO.

<table>
<thead>
<tr>
<th>Seq.</th>
<th>Monocular VO</th>
<th>Stereo VO</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ESP-VO</td>
<td>VISO2-M</td>
</tr>
<tr>
<td></td>
<td>( t_{rel}(\text{%}) )</td>
<td>( r_{rel}(\circ/100 \text{m}) )</td>
</tr>
<tr>
<td>03</td>
<td>6.72</td>
<td>2.76</td>
</tr>
<tr>
<td>04</td>
<td>6.33</td>
<td>0.52</td>
</tr>
<tr>
<td>05</td>
<td><strong>3.35</strong></td>
<td><strong>1.21</strong></td>
</tr>
<tr>
<td>06</td>
<td><strong>7.24</strong></td>
<td><strong>1.50</strong></td>
</tr>
<tr>
<td>07</td>
<td>3.52</td>
<td>1.71</td>
</tr>
<tr>
<td>10</td>
<td><strong>9.77</strong></td>
<td><strong>2.04</strong></td>
</tr>
</tbody>
</table>
| mean | **6.15**     | **1.63**   | 17.48     | 2.70      | 30.01     | 6.14     | 1.89 | 0.50

- \( t_{rel} \): average translational RMSE drift (\%) on length of 100m-800m.
- \( r_{rel} \): average rotational RMSE drift (\%/100m) on length of 100m-800m.
- The ESP-VO model used is trained on Sequence 00, 01, 02, 08 and 09. Its performance is expected to improve when it is trained on more data.
the translation. This may be due to the fact that the KITTI
dataset was recorded during car driving, which tends to
go straight on high speeds yet rotate when slowing down.
Moving forward, as a dynamics without significant changes
on rotation, can be easily modelled by the RNN in terms of
orientation. As the length of the trajectory increases,
the errors of both the translation and rotation of the ESP-
VO significantly decrease, approaching the stereo VISO2
as shown in Figure 9(a) and Figure 9(b).

The estimated VO trajectories of the testing sequences
corresponding to the previous experiment are given in
Figure 10 and Figure 11. It can be seen that the ESP-
VO produces relatively accurate and consistent trajectories
against the ground truth although it drifts over time,
demonstrating that the scale can be estimated with high
precision. We stress that neither scale estimation nor post
alignment to ground truth is performed for the ESP-VO
to obtain the absolute poses. The scale is completely
maintained by the neural network itself and implicitly
learned during the end-to-end training. On the other hand,
it is also shown that the results of both VISO2 and ORB-
SLAM suffer from serious scale drifts, which explains
the large errors in Figure 9. Take localisation results of
Sequence 05 in Figure 10 as an example. In Figure 10(b),
different segments of the trajectory of ORB-SLAM exhibits
big differences on the scales because the SLAM version
of the ORB-SLAM relies on global loop-closure detection
to correct scale drifts and reconstruct accurate, consistent
 trajectories (Strasdat et al. 2010). This can also be seen
in Figure 12 of the original ORB-SLAM paper (Mur-
Artal et al. 2015), where the result of Sequence 08 has
a big scale drift when no loop closure is detected and
the scale is not corrected. As for the monocular VISO2,
the localisation result in Figure 10(c) suggests that scale
estimation using fixed camera height is not robust to noise
due to car jolts during driving. Moreover, utilising camera
height is inapplicable to some scenarios, such as flying
robots or human walking. Since recovering accurate and
robust scale for monocular VO is surprisingly difficult, it
suggests an appealing feature of the DL based VO method
to estimate scale by exploiting prior knowledge learned
during training. The detailed performance of the algorithms
on the testing sequences is summarised in Table 2. It
indicates that ESP-VO achieves more robust results than the
monocular VISO2 and ORB-SLAM. Note that the previous
quantitative evaluation is based on the model trained only
on Sequence 00, 01, 02, 08 and 09 and, similar to most
of DL based methods, the performance of the ESP-VO is
expected to be significantly improved when it is trained on
more data. This is one of the major differences between
geometry based methods and learning based methods.

Although the generalisation of the ESP-VO model has
been evaluated in the previous experiment, in order to
 further investigate how it performs in totally new scenarios
with different motion patterns and scenes, the network
is tested on the testing dataset of KITTI VO/SLAM
benchmark. The ESP-VO model is trained on all the 11
training sequences of the KITTI VO benchmark (i.e.,
Sequence 00-10), providing more data to avoid overfitting
and maximise the performance of the network. Due to
the lack of ground truth for these testing sequences, no
quantitative analysis can be performed on the VO results.
For qualitative comparison, some predicted trajectories of
the ESP-VO, the VO ORB-SLAM, the monocular VISO2
and the stereo VISO2 are shown in Figure 12. It can be
seen that the ESP-VO outperforms the monocular VISO2
and ORB-SLAM, and it is the most similar one to the
stereo VISO2. It also seems that this larger training dataset
boosts the performance of the ESP-VO. Taking the stereo
properties of the stereo VISO2 into consideration, the ESP-
VO, as a monocular VO algorithm, achieves competitive
performance and generalises well in different unknown
scenarios. An exception could be the test on Sequence 12
in Figure 12(b) which suffers from rather high localisation
errors although the shape of the trajectory is close to the
stereo VISO2's. There are several reasons. First, the
training dataset does not have enough data on high speeds.
Among all the 11 training dataset, only the Sequence 01 has
velocities that are higher than 60 km/h. However, the speeds
of the Sequence 12 span from 50km/h up to about 90km/h.
Moreover, the images are captured at only 10 Hz, which
makes the VO estimation more challenging during fast
movement. The large open area around highway (lacking
of features) and dynamic moving objects, shown in Figure
14, can degrade the accuracy as well. These reasons also
apply to the Sequence 21. In order to mitigate these issues,
the conventional geometry based methods could increase
feature matching and introduce outlier rejection, such as
RANSAC. However, for the DL based method, it is unclear
how to embed these techniques yet. A feasible solution is
to train the network with more data which not only reflects
these situations but also is deliberatively augmented with
noises, outliers, etc., allowing the network to figure out how
to deal with these problems.

Some VO results on the KITTI raw data compared with
GPS/INS ground truth are given in Figure 13 with 5 sample
images of each sequence. It can be seen that, similar to the
KITTI VO/SLAM testing image sequences, the images are
captured in various environments with pedestrians, cyclists,
cars under different lighting conditions, shadows, etc.
The proposed ESP-VO demonstrates strong generalization
capabilities to the challenging image sequences.

6.5.2  Testing results on Cityscapes and Málaga
datasets
In order to further evaluate the generalisation of the EPS-
VO on different platforms in new environments, Cityscapes
Fig. 12. Trajectories of VO results on the testing Sequence 11, 12, 15, 17, 18 and 19 of the KITTI VO benchmark (no ground truth is available for these testing sequences). The ESP-VO model used is trained on the whole training dataset (00-10) of the KITTI VO benchmark and its scales are recovered automatically from neural network without alignment to ground truth.

and Málaga datasets are used to directly test the model trained on KITTI dataset. Note that all the results in this part are produced by directly testing on the Málaga or Cityscapes datasets without any training or fine-tuning on them.

Fig. 15 and Fig. 16 show the testing results on the Cityscapes (Stuttgart_00 and Stuttgart_01 sequences) and
Fig. 13. Trajectories of ESP-VO testing results on Sequence 39, 61, 64, 95 of KITTI raw dataset with 5 sample images for each sequence.

The experiment further verifies that the ESP-VO is able to generalise although the datasets are collected with different devices (e.g., cameras and cars) in totally fresh environments showing different city scenarios (see the corresponding sample images of the datasets in Fig. 7).

6.6 Results in MAV scenario

EuRoC dataset is used to evaluate the performance of the proposed ESP-VO method for complex flying motion. Since EuRoC dataset only has 5 image sequences in the machine hall, we train the neural networks with different combination of sequences. Specifically, 4 of Machine Hall sequences are used for training a neural network and the trained model is tested on the remaining sequence, which means for each training there are only about 10 minutes image sequences. Since this is very small number of training data for the DL based ESP-VO, the experimental results reflect its performance when trained only with limited data.

Figure 17 shows the results of the ESP-VO (model trained on Machine Hall 01, 02, 03 and 05) on the testing sequence, Machine Hall 04 ("difficult"), with a sample image. We can see from the sample image that some part of the environment is very dark, making the images under-exposed. The ESP-VO is able to recover the shape and scale of the trajectory accurately although it drifts over...
Fig. 15. Testing results on Cityscapes dataset without training or fine-tuning on it. The ESP-VO used is only trained on the training dataset (00-10) of the KITTI.

time. Note that there are very few training images (only some in Machine Hall 05) that are taken in this dark area. Surprisingly, the network learns how to cope with these under-exposure images by using this small number of training data. The corresponding 6 DoF position and orientation estimation is given in Figure 17(b) against ground truth. It can be seen that the network tracks changes on position and orientation well, which means the dynamics of the flying robot is learned by the network. However, since the dynamics of a flying robot are more complex than that of a driving car, the network requires more diverse training data in order to thoroughly learn the motions. Unfortunately, the data of the EuRoC dataset is limited and there is no public available large-scale dataset of a flying robot which can be used to train a DNN for VO estimation. The results of Machine Hall 05 (“difficult”) in Figure 18 suggest similar findings to Machine Hall 04.

Fig. 16. Testing results on Málaga dataset overlaid on Google maps without training or fine-tuning on it. The ESP-VO used is only trained on the training dataset (00-10) of the KITTI.
6.7 Results in indoor motion scenario

The experiments on the indoor motion scenario are conducted as two parts. First, we use our self-collected dataset to train and test ESP-VO models. Then, we directly use the NYU dataset for testing without any further training or fine-tuning.

6.7.1 Results on self-collected dataset

The proposed ESP-VO method is trained and tested in large-scale indoor environments based on the indoor motion dataset. The network model is trained on the training data which was collected in a departmental building, and then tested in another office building and a museum.

The testing result of the proposed method in an office building is shown in Figure 19 along with some sample images. It can be seen that the dataset is very challenging for monocular VO because the images are captured under different lighting conditions and some of them mostly contain texture-less white walls in narrow corridors. Fast motion, like the turning shown in the consecutive images in Figure 19, also causes significant scene changes between frames, which makes the dataset more difficult for VO. Nevertheless, the ESP-VO still tracks the shape of the ground truth trajectory (Tango with SLAM) and reconstructs the whole trajectory although it drifts over...
Wang et al.

Fig. 20. Course of losing tracking of ORB-SLAM with images to show features being tracked.

Fig. 21. Sample images and result of ORB-SLAM in a museum with a busy Café. Images show challenging lighting conditions and walking people. Big scale drift appears on different segments of the ORB-SLAM trajectory.

Fig. 22. Course of losing tracking of LSD-SLAM with colour-coded depth map and continuous image sequence.

Fig. 23. Localisation results on the museum dataset.

time. We also attempted to run ORB-SLAM and LSD-SLAM on this dataset but both of them either refused to initialise or lost tracking easily and could not finish localisation. More specifically, a course of losing tracking of ORB-SLAM is given in Figure 20. It indicates that when most parts of an image frame are low-texture white wall, the number of feature correspondences between a current image frame and a keyframe decreases dramatically before losing tracking. This is a well-known limitation of feature based methods. Moreover, the feature based methods are prone to get lost during fast motion which produces substantial scene changes between image frames. These also explain why LSD-SLAM lost tracking. The experiment suggests that DL based VO method can be more robust to low texture environments and agile movement after being trained with similar scenarios. This proposes a promising direction of utilising DL based approach as a remedy when conventional geometry based methods lose tracking or fail. Note that the Project Tango device works well in this scenario because it relies on IMU and high-frame wide-angle fish-eye camera to conduct visual inertial localisation with significantly increased view and reduced scene changes. As for the ESP-VO, it only uses normal perspective images at 2 Hz since most robots and mobile devices are only equipped with normal cameras with limited frame rate.

The VO algorithms are also tested in a public museum to further evaluate its performance in new, dynamic environments. Figure 21 illustrates the localisation result and feature map of ORB-SLAM with some sample images showing challenging lighting conditions and walking people around. It can be seen that there are big scale drifts on different segments of the ORB-SLAM trajectory because of an extra number of gauge degrees of freedom of monocular VO (Strasdat et al. 2010). Since no global loop exists on this trajectory (see whole trajectory in Figure 23), even the SLAM version of ORB-SLAM with scale drift-aware correction is not able to mitigate this problem. In terms of LSD-SLAM, it loses tracking during turning as shown in Figure 22. This is because tracking poses during fast rotation is challenging by using low-rate image frames, and automatic exposure changes may violate the brightness
constancy assumption of direct methods. The localisation results of the VO algorithms and Project Tango are given in Figure 23. It demonstrates that the ESP-VO can estimate VO accurately by only using monocular vision without explicit scale estimation or post alignment to ground truth. The ORB-SLAM result is manually aligned to ground truth with the starting point as origin. The ESP-VO’s estimates on position (x, y and z) and orientation (roll, pitch and yaw) are given in Figure 24 compared with these of Project Tango. It shows that the ESP-VO predicts both position and orientation with a reasonable precision.

6.7.2 Testing results on NYU depth dataset

We use the NYU depth dataset to directly test ESP-VO model trained on our self-collected dataset, evaluating its generalisation in completely new scenarios.

As discussed before, there is no ground truth pose available in the NYU dataset for evaluation purpose. Because of the difficulties caused by relatively narrow field of view of Kinect camera, limited texture in indoor environments and fast camera motion, NYU dataset is seldom employed to test monocular VO and SLAM (according to our test the ORB-SLAM and LSD-SLAM often refuse to initialise or lose tracking). Therefore, the RGBD-SLAM (original version with loop closure detection and graph optimisation) (Endres et al. 2014) is adopted to produce motion trajectories for comparison. Note that apart from RGB images RGBD-SLAM uses Kinect’s depth images, which furnish valuable depth information to significantly benefit pose estimation and SLAM. In contrast, ESP-VO only utilises monocular RGB imagery.

Fig. 25 shows the results of ESP-VO and RGBD-SLAM on NYU nyu_office_0 sequence along with two consecutive images. Note the big change on consecutive images at Location A and B. RGBD-SLAM can estimate the poses with high precision thanks to the depth information and map maintained during SLAM although it still has some errors after SLAM back-end optimisation. Compared with the localisation result of the RGBD-SLAM, ESP-VO as a monocular VO technique recovers the motion trajectory with reasonable accuracy although it suffers from relatively big drifts at Location A, B and C. According to the consecutive images in Fig. 25(b), we can see that it is very challenging for a VO method to work well at Location A and B due to the lack of image overlap caused by severe angular motion. In particular, Location B presents an extreme case whose pair of consecutive images shares no overlap at all, leading to failure at feature matching for feature based VO algorithms. This usually happens in indoor environments with fast motion and limited field of view.

We further test the ESP-VO with NYU bedroom_0002 sequence. Because the ESP-VO is trained with data collected solely in office buildings, it has not “seen” any bedroom before. This means the style of the testing environment is totally new to the ESP-VO model. Localisation results of ESP-VO and RGBD-SLAM on this sequence along with 10 sample images are presented in Fig. 26. We can see that ESP-VO continuously tracks the motion in this unseen environment, while RGBD-SLAM
encounters big sudden jerks on estimated locations during passing the narrow corridor (image 7 to 9 in Fig. 26).

Since the NYU dataset is not used for training or fine-tuning the ESP-VO at all, the generalisation of the ESP-VO in new scenarios is validated again.

6.8 Uncertainty estimation

The uncertainty estimation of the proposed ESP-VO is evaluated in this section. We take KITTI dataset as an example here. In order to have ground truth to evaluate covariance estimation, the ESP-VO model employed is trained on Sequence 00, 01, 02, 08 and 09, leaving other sequences for testing. The testing results on Sequence 06 are analysed in detail.

Errors on VO estimates between two consecutive images against their $3\sigma$ covariance intervals are shown in Figure 27. It can be seen that the errors on position and orientation are all located between the covariance intervals, which verifies the meaningful uncertainty estimation from the ESP-VO. We can also see that the neural network captures the two turnings in Sequence 06 (see the trajectory in Figure 29) and reflects the uncertainty on the estimated covariance. This is very interesting since this ability is automatically learned by the neural network during training pose estimation with no supervision on uncertainty (no label for uncertainty estimation in Section 4). The corresponding covariances of the whole trajectory on x, y and yaw are given in Figure 28. It indicates that the VO drifts are consistently bounded by the $3\sigma$ covariance intervals, and the yaw drift increases incrementally and more on turning. The VO trajectory and ground truth on x-y (for clarity only show in 2D ) are presented in Figure 29 along with covariance ellipses. It can be seen that although the trajectory drifts away, its covariance ellipses continuously cover the corresponding ground truth poses. This analysis verifies the effectiveness of the ESP-VO on uncertainty estimation by using end-to-end DL. Since the ESP-VO is only a VO estimation method, similar to dead-reckoning, there is no Bayes filter based update phase, which means we cannot perform consistency check using innovation or normalised estimation error squared (NEES) (Bar-Shalom et al. 2004).

In order to further evaluate the covariances estimated from the ESP-VO, its VO is extended for robot localisation based on pose graph SLAM. Since covariances of odometry edges are essential in pose graph SLAM, it is accurate to analyse the performance of the covariance estimation in the context of pose graph SLAM. In order to perform SLAM, DLoopDetector (Gálvez-López and Tardós 2012), which detects loops by combining bag-of-words with temporal and geometrical constraints, is employed for loop closure detection. Three pose graphs of Sequence...
06 are built, one of which uses the covariances from the ESP-VO while the other two are based on two different fixed covariances. After building the pose graphs, g2o (Kümmerle et al. 2011) is used for optimisation with 30 iterations to make sure they all converge. The pose graph SLAM results on Sequence 06 are shown in Figure 30. Although most of current VO and visual SLAM methods assume a constant covariance of odometry prediction for simplicity, it can be seen from the figure that a bad covariance on VO can seriously degrade the localisation accuracy of the pose graph SLAM. The trajectory produced by using the covariances calculated from the ESP-VO is more accurate than the ones using constant covariances. This is because ESP-VO computes a specific uncertainty for each of VO estimates according to the raw images and the dynamic models learned. We emphasise that these three pose graphs are built and optimised with identical VO estimates, loop closures (both transformation and covariance), graph SLAM optimiser, iteration steps, etc. The only difference between them is the covariances on VO. The pose graph SLAM results on KITTI Sequence 05 and 07 (only these two as well as 06 have loop closures in the testing sequences 03, 04, 05, 06, 07 and 10) using VO and covariance estimated from the ESP-VO are also presented in Figure 29. It can be seen that by using the covariances from the ESP-VO, we can achieve accurate localisation results based on pose graph SLAM, validating its capability on uncertainty estimation. Because estimating covariance from visual information involves many non-linearities and uncertainties, it is very tricky to derive consistent and accurate uncertainty estimates. Therefore, this experiment suggests that DL based uncertainty estimation could be an appealing approach.

Note that the pose graph SLAM experiments conducted here are not dedicated to solving the monocular visual SLAM problem or building an accurate monocular SLAM system. In fact, they are performed to evaluate the performance of the ESP-VO’s uncertainty estimation instead. No advanced technique, such as optimisation on group of similarity transformation $\text{Sim}(3)$, local BA, etc., is adopted here.

6.9 Runtime, reliability and limitations

Since real-time operation is critical for robotic applications and Deep Learning based methods are generally considered to be slow, here we discuss the real-time performance of the ESP-VO. A NVIDIA Tesla K40 GPU and a MacBook Pro laptop (2.8 GHz CPU and 16GB RAM) are used.
to compute the runtime of online inference on GPU and CPU, respectively. Because in practice images are captured one by one over time, the runtimes presented here are computed online as per-frame, i.e., the batch size of the neural network is 1, including all image pre-processing, e.g., image reading and resizing. Histograms of per-frame runtime (image size 320×240) runtime in second on both GPU and CPU are shown in Fig. 32. It can be seen that per-frame runtime for each prediction is between 40 ms and 55 ms on the GPU, while between 150 ms and 180 ms on the CPU. The average per-frame running time is about 46.8 ms and 164.7 ms on GPU and CPU, respectively. Therefore, ESP-VO is capable of running at about 20 Frames Per Second (FPS) on GPU and 6 FPS on CPU. Note that compared to the online inference discussed here, offline inference which can load multiple images (batch size bigger than 1) at each time is usually much faster thanks to parallel computing, especially on GPU.

The proposed ESP-VO is trained and tested all with real world data exhibiting various challenging lighting conditions, image over-exposure and under-exposure, image blur, dynamic objects, low-texture surroundings, agile motion, fast rotation, etc., in practice. Suffering from these, the conventional geometry based methods may refuse to initialise, lose tracking and increase scale drifts. During our testing, we have not encountered problems, e.g., losing tracking or declining to produce VO results, for the ESP-VO. An example video* of testing the ESP-VO with a mobile phone camera (Google Nexus 5 smartphone) in a supermarket demonstrates this. Since the images suffer from very serious rolling-shutter effect and blur (see sample images in Fig. 33 but best to watch the video), this testing video is considered to be very challenging for traditional VO approaches. In contrast, the ESP-VO can work well. This is because given an input, the neural network always produces prediction.

However, it has two sides, e.g., it may incur big errors for wrong or “unseen” input. Therefore, the key problem of DL based method is how to improve generalisation. In our experiments, the generalisation of ESP-VO has been validated by directly testing it on new datasets which are collected with different devices in totally new environments. Because the features learned by the CNN are geometric ones related to optical flow rather than appearance, it can generalise to new scenarios without fine tuning. Similar to all supervised learning based approaches, the performance of the DL based method is highly determined by the quality of the training dataset and the similarity (in terms of feature representation rather than image appearance) shared between the training and testing data. Hence, large-scale dataset usually improves the generalisation and results of the trained models. For example, the result on Sequence 21 of the KITTI dataset in Figure 34 shows big errors of the ESP-VO compared to the stereo VISO2. Since this sequence was captured at high speed on highway with large open areas and there is limited training data in KITTI reflecting this scenario, the neural network does not have high performance. To enhance its

*ESP-VO in supermarket: https://youtu.be/M4v_-XyYKHY
Fig. 33. Sample images of a rolling-shutter mobile phone camera used to test the ESP-VO in a supermarket. It is recommended to watch the video to see the serious rolling-shutter effect.

Fig. 34. Results on KITTI Sequence 21 with sample images. The sequence was taken at high speeds on a highway with many dynamic driving cars. No ground truth is available.

reliability, it needs to be trained with substantial data similar to this situation.

The scale of ESP-VO is implicitly learned during training without using extra information or device. The experimental results on directly testing models trained on KITTI dataset and Human motion dataset on Cityscapes and Málaga datasets and NYU depth dataset show that our models can estimate the scale accurately across different environments. However, training a model in a single scenario may lead to reduced performance on it. Therefore, it is necessary to train a model cross different scenarios for scale estimation.

7 Discussion and open questions

In this section, we discuss some problems of VO and potentially how to improve it. Some open questions for future research on DL based VO and visual SLAM are also proposed.

7.1 Learning to be more accurate and robust

The robotics and computer vision communities have been working on VO for several decades, increasingly garnering valuable knowledge and deep understanding of VO systems. Meanwhile, the state-of-the-art algorithms become more accurate and computationally efficient in larger environments, enabling many applications on autonomous driving, augmented reality, virtual reality, etc. However, when considering robustness, we still encounter failures, such as refusing to initialise and losing tracking, quite often during operation in challenging scenarios, e.g., low-texture corridors and fast motion. Although we usually evaluate VO and visual SLAM methods in terms of accuracy, we argue that in some cases robustness is more important. An accurate yet fragile VO system, for instance, is not as appealing or practical as a reliable one with reasonable accuracy in the context of self-driving cars.

Currently, we still need to spend plenty of time in manually analysing the failure cases and then patching up our algorithm when engineering a VO or visual SLAM system. This may be repeated hundreds of times before achieving superior performance. However, a question is what the algorithm learned and benefited from this repeated procedure. The answer perhaps is some of our knowledge and skills we want it to master. There is nothing wrong with this, but it may only be able to progress incrementally and slowly. Therefore, would it be possible to accelerate this? Recent successful stories of DL, such as the superhuman accuracy of ImageNet Challenge, suggest an alternative way to improve VO systems. Since we have already armed with some excellent VO and visual SLAM algorithms, we may gain significantly more if we guide them to automatically learn how to deal with the challenging scenarios by using real-world data.

Every method has its own pros and cons. The DL based monocular VO method proposed can be mostly competitive to the state-of-the-art monocular VO algorithms and, in some situations, even stereo VO. However, based on recently emerging DL technique, it also has many questions to be solved. Generalisation is a key one since the DL based algorithm works well provided that the training and testing data shares a certain similarity. For similarity, here we mean similarity in terms of feature representation and dynamic models learned by the DNN rather than naive image appearance. For example, it would be difficult to train a model by using all data collected outdoor from a driving car, and then directly testing it indoor on flying robots if
their data exhibits no similarity. The DNN needs to be fine-
tuned or jointly trained if it is expected to work in both
situations.

7.2 What may facilitate DL in VO and robotics

Although DL has revolutionised computer vision in recent
several years, it has not been popular in robotics yet. There
are two aspects that we can think of.

First is the data available to train DNNs for various
tasks. Although it is known that a large amount of data
is required to train DNNs, there are limited number of
large-scale high-quality datasets available in robotics. In
terms of VO and visual SLAM, it is luck to have high-
quality public datasets, like KITTI and EuRoC, for training
DNNs. However, since most of them are generated for
developing and testing conventional geometry based VO
and visual SLAM methods, they are not really large-scale
from the perspective of DL. For example, the training data
of KITTI (Sequence 00-10) has about 23k images recorded
in 38 minutes during 20 km travel, while the ImageNet
dataset in computer vision has over ten million manually
annotated images for object recognition. Therefore, large-

scale, high-quality, public datasets are demanded to push
the DL research in robotics, like ImageNet did for object
classification and MS COCO did for image captioning. It is
exciting that this kind of datasets have been gradually made
in robotics, such as the large-scale robotic grasping dataset
in (Levine et al. 2016).

The other is the architecture of the DNN. Since
many computer vision tasks are performed on a single
static image, e.g., object recognition, CNNs are powerful
enough to learn feature representation for solving them.
Therefore, most of current works on computer vision
are based on CNNs. However, dealing with single image
using CNNs is inadequate and less effective for most of
problems in robotics because robotic systems usually have
temporal structures and inherently process sequential data.
Therefore, RNNs, which can learn dependencies and model

complex dynamics from sequential data, are incredible
useful and more powerful in robotics although currently
they are mainly used in speech recognition, machine
translation, etc. Since RNNs can be trained in an end-to-
end manner, many connections and dependencies which
are difficult to be explicitly modelled or utilised by hand-
crafted design can be learned. Moreover, RNNs are able to
handle sequential data with variable lengths, making it ideal
for robotic systems that have no fixed length of input. A
problem of RNNs is that they are not as effective as CNNs
in terms of extracting features from high-dimensional data,
e.g., raw images. Therefore, in some cases, we need DNNs
which have the capabilities of both feature learning and
sequential modelling, such as the RCNN used in this work.

7.3 Open questions on DL based VO and
visual SLAM

The proposed ESP-VO is only a starting step of DL based
VO and visual SLAM. There are still many open problems
which need to be tackled in order to move towards to a DL
based truly efficient and robust VO or visual SLAM system.

7.3.1 Incorporation with geometry based methods

As previously discussed, it is essential to incorporate DL
based methods with conventional geometry based methods
for VO estimation. However, it is not clear in which way
this should be realised to be most effective.

7.3.2 Combining loop closure detection for SLAM

In order to correct drift of a VO system over time, one
solution is to perform SLAM by detecting loop closures.

We have shown some preliminary results on pose graph
SLAM using results from the ESP-VO. But it is unclear
whether it is ideal to achieve this by using a traditional
framework, e.g., graph SLAM, or by designing a new end-
to-end paradigm of SLAM based on DL.

7.3.3 Unsupervised learning

The DNNs of the proposed DL based method is
trained by using supervised learning for VO estimation,
which remains an obstacle to benefitting from large-scale
unlabelled data in practice. Enabling unsupervised learning
should provide substantially more data to improve the
performance of the DL based method. A potential solution
would be using loss functions based on geometry, as in
(Garg et al. 2016) and (Kendall and Cipolla 2017).

7.3.4 Reinforced system

We cannot build a perfect VO system in one stroke, then
it would be remarkable to make it gradually improve during
every single test or usage. Deep reinforcement learning may
be able to define a good policy to guide the VO system
towards to this.

8 Conclusion

This paper presents a novel end-to-end, sequence-to-
sequence probabilistic monocular VO algorithm based on
Deep Learning, demonstrating how to make full use of
the data we may have already collected to build and
improve a VO system. Leveraging the power of deep
RCNNs, this new paradigm is able to automatically learn
both feature representation and sequential models for
monocular VO from a sequence of raw images. Since no
module of the conventional VO algorithms (even camera
calibration) is adopted and the DNNs are trained in an
end-to-end manner, there is no need to carefully tune the
parameters of the VO system. Based on the extensive
experiments on three different datasets, which are collected
during outdoor car driving, indoor robot flying and human
walking, it is verified that the proposed monocular VO method, ESP-VO, can produce competitive VO results with meaningful uncertainties and precise scales in completely new, challenging scenarios.

Although the proposed DL based VO method shows some potential in this area, we stress that it is not expected as a replacement to the classic geometry based approaches. On the contrary, it is would be a viable complement, i.e., incorporating geometry based approaches with the representation, knowledge and models learned by the DNNs to further improve the VO systems in terms of accuracy and, more importantly, robustness. Based on the DL method, future work will focus on how to incorporate geometry based methods and increasingly improve the performance of a VO system during daily usage.

Acknowledgements

The authors would like to acknowledge the use of the University of Oxford Advanced Research Computing (ARC) facility in carrying out this work (http://dx.doi.org/10.5281/zenodo.22558).

Funding

This work is supported by EPSRC Programme Grant “Mobile Robotics: Enabling a Pervasive Technology of the Future” (EP/M019918/1).

References


239–248.
Strasdat H (2012) Local accuracy and global consistency for efficient visual SLAM. PhD dissertation, Imperial College London.