

DLR-IB-RM-OP-2020-82

**Uncertainty-Aware Attention
Guided Sensor Fusion For
Monocular Visual Inertial Odometry**

Masterarbeit

Kashmira Shinde



DLR

**Deutsches Zentrum
für Luft- und Raumfahrt**

MASTERARBEIT

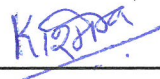
UNCERTAINTY-AWARE ATTENTION GUIDED SENSOR FUSION FOR MONOCULAR VISUAL INERTIAL ODOMETRY

Freigabe:

Der Bearbeiter:

Unterschriften

Kashmira Shinde



Betreuer:

Jongseok Lee



Der Institutsdirektor

Prof. Alin Albu-Schäffer

alin.albu-sch
aeffer@dlr.de

Digital signiert von
alin.albu-schaeffer@dlr.de
DN: CN=alin.albu-schaeffer@dlr.de
Grund: Ich bin der Verfasser dieses
Dokuments
Ort: hier den Ort der Signierung ein
Datum: 2020.10.30 19:25:56+01'00'
Foxit PhantomPDF Version: 10.1.0

Dieser Bericht enthält 61 Seiten, 31 Abbildungen und 6 Tabellen

Uncertainty-Aware Attention Guided Sensor Fusion for Monocular Visual-Inertial Odometry

Kashmira Shinde

Master's Thesis – June 02, 2020.

Automation & Robotics
Faculty of Electrical Engineering and Information Technology
Technical University of Dortmund

in cooperation with
Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR)
and the Ruhr-Universität Bochum (RUB).

1st Supervisor: Prof. Dr.-Ing. Aydin Sezgin (RUB)
2nd Supervisor: M.Sc. Jongseok Lee (DLR)



Deutsches Zentrum
für Luft- und Raumfahrt
German Aerospace Center

RUHR
UNIVERSITÄT
BOCHUM

RUB

Acknowledgements

This work has been carried out in the year 2019/20 at the German Aerospace Center (DLR), at the Institute of Robotics and Mechatronics. I would like to thank my supervisor at the DLR, Jongseok Lee for his continuous guidance, support and granted freedom. Furthermore, I would like to thank my university supervisor Prof. Dr.-Ing. Aydin Sezgin, Head of Chair for Digital Communication Systems at Ruhr University of Bochum, for the supervision, advice and smooth execution of my thesis. I am grateful to Prof. Dr.-Ing. Alin Albu-Schäffer, director of the Institute of Robotics and Mechatronics at the DLR, for giving me the opportunity to carry out this work in the department of Perception and Cognition. Furthermore I want to thank my colleagues at DLR, especially Matthias Humt, for the support and valuable discussions. Finally, I would like to express my heartfelt gratitude to my family for their unconditional love and unwavering support and encouragement at every step of my life.

Abstract

Visual-Inertial Odometry (VIO) refers to dead reckoning based navigation integrating visual and inertial data. With the advent of deep learning (DL), a lot of research has been done in this realm yielding competitive performances. DL based VIO approaches usually adopt a sensor fusion strategy which can have varying levels of intricacy. However, sensor data can suffer from corruptions and missing frames and is therefore imperfect. Hence, need arises for a strategy which not only fuses sensor data but also selects the features based on their reliability.

This work addresses the monocular VIO problem with a more representative sensor fusion strategy involving attention mechanism. The proposed framework neither needs extrinsic sensor calibration nor the knowledge of intrinsic inertial measurement unit (IMU) parameters. The network, being trained in an end-to-end fashion, is assessed with various types of sensory data corruptions and compared against popular baselines. The work highlights the complementary nature of the employed sensors in such scenarios. The proposed approach has achieved state-of-the-art results showing competitive performance against the baselines, thereby contributing to an advance in the field. We also make use of Bayesian uncertainty in order to obtain information about model's certainty in its predictions. The model is cast into a Bayesian Neural Network (BNN) without making any explicit changes in it and inference is made using a simple tractable approach - Laplace approximation. We show that notion of uncertainty can be exploited for VIO and sensor fusion, particularly that sensor degradation results in more uncertain predictions and the uncertainty correlates well with pose errors.

Eidesstattliche Versicherung (Affidavit)

Shinde, Kashmira

Name, Vorname
(Last name, first name)

207184

Matrikelnr.
(Enrollment number)

Ich versichere hiermit an Eides statt, dass ich die vorliegende ~~Bachelorarbeit~~/Masterarbeit* mit dem folgenden Titel selbstständig und ohne unzulässige fremde Hilfe erbracht habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie wörtliche und sinngemäße Zitate kenntlich gemacht. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

I declare in lieu of oath that I have completed the present ~~Bachelor's~~/Master's* thesis with the following title independently and without any unauthorized assistance. I have not used any other sources or aids than the ones listed and have documented quotations and paraphrases as such. The thesis in its current or similar version has not been submitted to an auditing institution.

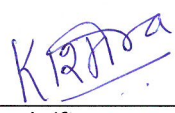
Titel der ~~Bachelor~~-/Masterarbeit*:
(Title of the ~~Bachelor's~~/ Master's* thesis):

Uncertainty-Aware Attention Guided Sensor Fusion for Monocular Visual-Inertial Odometry

*Nichtzutreffendes bitte streichen
(Please choose the appropriate)

Dortmund, 02.06.2020

Ort, Datum
(Place, date)


Unterschrift
(Signature)

Belehrung:

Wer vorsätzlich gegen eine die Täuschung über Prüfungsleistungen betreffende Regelung einer Hochschulprüfungsordnung verstößt, handelt ordnungswidrig. Die Ordnungswidrigkeit kann mit einer Geldbuße von bis zu 50.000,00 € geahndet werden. Zuständige Verwaltungsbehörde für die Verfolgung und Ahndung von Ordnungswidrigkeiten ist der Kanzler/die Kanzlerin der Technischen Universität Dortmund. Im Falle eines mehrfachen oder sonstigen schwerwiegenden Täuschungsversuches kann der Prüfling zudem exmatrikuliert werden. (§ 63 Abs. 5 Hochschulgesetz - HG -).

Die Abgabe einer falschen Versicherung an Eides statt wird mit Freiheitsstrafe bis zu 3 Jahren oder mit Geldstrafe bestraft.

Die Technische Universität Dortmund wird ggf. elektronische Vergleichswerkzeuge (wie z.B. die Software „turnitin“) zur Überprüfung von Ordnungswidrigkeiten in Prüfungsverfahren nutzen.

Die oben stehende Belehrung habe ich zur Kenntnis genommen:

Official notification:

Any person who intentionally breaches any regulation of university examination regulations relating to deception in examination performance is acting improperly. This offense can be punished with a fine of up to €50,000.00. The competent administrative authority for the pursuit and prosecution of offenses of this type is the chancellor of TU Dortmund University. In the case of multiple or other serious attempts at deception, the examinee can also be unenrolled, section 63, subsection 5 of the North Rhine-Westphalia Higher Education Act (*Hochschulgesetz*).

The submission of a false affidavit will be punished with a prison sentence of up to three years or a fine.

As may be necessary, TU Dortmund will make use of electronic plagiarism-prevention tools (e.g. the "turnitin" service) in order to monitor violations during the examination procedures.

I have taken note of the above official notification:**

Dortmund, 02.06.2020

Ort, Datum
(Place, date)


Unterschrift
(Signature)

****Please be aware that solely the German version of the affidavit ("Eidesstattliche Versicherung") for the Bachelor's/ Master's thesis is the official and legally binding version.**

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Related Work	2
1.3	Contribution and Outline	4
2	Preliminaries	7
2.1	Basic building blocks	7
2.1.1	Convolutional Neural Network	7
2.1.2	FlowNet	7
2.1.3	Recurrent Neural Networks	8
2.1.4	Long Short-Term Memory	9
2.2	Fusion methods	10
2.2.1	Early Fusion	10
2.2.2	Late Fusion	10
2.2.3	Intermediate Fusion	11
2.3	Self-attention mechanisms	11
2.3.1	Self-attention	11
2.3.2	Attention variants	12
2.4	Probabilistic Machine Learning	12
2.4.1	Bayesian Neural Networks	13
2.4.2	Model uncertainty	13
2.4.3	Aleatoric uncertainty	13
2.4.4	Uncertainty aware attention mechanism	14
2.4.5	Laplace Approximation	14
3	Approach	15
3.1	End-to-end monocular VIO architecture	15
3.1.1	Feature extractors	16
3.1.2	Feature fusion mechanism	17
3.1.3	Sequential modelling and pose regression	17
3.1.4	Loss function	18
3.2	Multi-Head Self-Attention	18
3.3	Uncertainty estimation	19
3.3.1	Fisher information matrix	20

3.3.2	Parameters of Laplace method	20
3.3.3	Sampling weights	21
3.4	Uncertainty quantification	21
3.4.1	Mean Squared Error	21
3.4.2	Variance	22
3.5	Prediction reliability of neural networks	22
3.5.1	Degradation scenarios	22
4	Experiments, Results and Discussions	25
4.1	Baseline setup and implementation	25
4.1.1	Baselines	25
4.1.2	Dataset	26
4.1.3	Evaluation metric	27
4.2	Validation	27
4.2.1	Training and testing	27
4.2.2	Empirical analysis	28
4.3	Sensor fusion representation in VIO Soft	32
4.4	Analytical outlook on network expressivity	33
4.5	Bayesian framework	34
4.5.1	Curvature approximation and sampling	34
4.5.2	Hyperparameters of the BNN	34
4.5.3	Results for uncertainty estimates	34
5	Conclusion	37
	List of Figures	39
	List of Tables	41
	List of Formulas	43
	Bibliography	45

1 Introduction

1.1 Motivation

Aerial manipulation aims at performing manipulation tasks using robotic arm mounted on an agile aerial platform[54]. Control of the aerial manipulator remotely with the help of an operator leads to aerial tele-manipulation. Recently, the world’s first cable-suspended aerial telepresence system has been successfully developed at our institute and applied to maintenance and inspection tasks, in which a robotic system called SAM (Suspended Aerial Manipulator) deployed and retrieved a pipe inspection robot[39]. Fig. 1.1 depicts SAM. During our endeavors, we identified a new functional requirement of a perception module, that is, to provide a 3D information of the scene to the teleoperator. This is because 1) the operator is not guaranteed to have a close and direct visual contact with the scene, and 2) a visual feedback solely based on the streams of 2D images is not sufficient to achieve a difficult aerial manipulation task. This could be achieved by using virtual reality and hence, in our previous work, we integrated this concept along with the use of visual-inertial odometry (VIO) to make the system more robust[39]. However, it was observed that 1) VIO might improve, 2) we might use simultaneous localisation and mapping (SLAM) for solving the same problem but it requires a good front-end or odometry. For this reason, the current work addresses this task of providing a better and improved odometry.

Various techniques like visual odometry (VO), VIO have been researched extensively over the past few years in the fields of robotics and computer vision for estimating egomotion, i.e. the three dimensional displacement of a sensor (for e.g. a camera) in an environment[40, 57, 41, 21, 52]. Cameras and IMU are the preferred choice of sensors as they offer several advantages like being lightweight, low cost and power efficient. A conventional pipeline is followed by existing methods for VIO which includes preprocessing sensor data, feature detection, tracking and a fusion framework. These classical methods are generally hard coded and are often fine tuned. Moreover, extrinsic sensor calibration, temporal alignment and scale estimation needs to be done explicitly. Learning based approaches making use of deep neural networks for VIO have showcased performance robustness, eliminating the need for these steps. However, these methods fail to take into account sensor corruptions which are very plausible in real life scenarios, rendering them difficult to use in safety critical applications.

Herein lies the need for a framework which gives importance to reliable features from one modality when the other degrades. We thus propose a generic end-to-end deep learning based monocular VIO approach using self-attention mechanism for feature selection. With the help of the employed multi-head self-attention mech-



Figure 1.1: Cable-suspended aerial manipulator SAM [55]

anism, the model learns to attend to parts of the combined visual-inertial feature space depending upon environmental dynamics. The results of our work highlight the complementary nature of the visual and inertial sensors - more preference is given to visual features for translational motion and to inertial features during rotation. Moreover, the network is incorporated with the ability to say - ‘I don’t know’, when it is uncertain about the predicted poses in presence of corrupted data.

1.2 Related Work

We now review some of the previous classical as well as deep learning solutions to the VO and VIO problem.

In traditional sparse feature-based methods for VO, a conventional pipeline is followed - sensor calibration, detection and tracking of features, motion estimation and optimization[57]. Feature based keyframe (a reference frame for each frame subsequence) methods such as ORB-SLAM[47] and PTAM[37] primarily use salient features to associate measurement with the geometric landmark. These works have been extensively used in robotics because they could give real-time performance and provide loop closure[47]. However, these methods require outlier rejection for feature matching and are generally sensitive to it. The other class of methods such as DTAM[50], DSO[18] and LSD-SLAM[19], called direct methods utilize the pixel information from entire image and minimize photometric errors. They perform better than their feature-based counterparts, in the sense that they are feature-less, less

noisy, work well in smooth environments and have less computational overhead. However, these SLAM methods require a good initial guess of pose for continuous regression of camera's pose.

Some inherent drawbacks in classical monocular VO are drift and scale ambiguity. Recent developments in the field of deep learning (DL) for visual odometry[63, 35, 11] have shown to better performance metrics such as accuracy and robustness. [63] describes an end-to-end trainable approach for VO. First, a Convolutional Neural Network (CNN) adapted from [16] estimates optical flow and learns geometric feature representation. Motion dynamics are then modelled by examining the connections between a sequence of images or rather CNN features by Long Short-Term Memory (LSTM) and finally pose regression is done to give the estimated poses. One of the first deep learning approaches that used CNNs for estimation of 6 degrees of freedom poses is PoseNet[35]. Input to the network is single RGB and transfer learning is used in first stage. PoseNet showed a better performance than SIFT-based methods on images with smooth texture-less regions. Some works have shown improved performance by making certain modifications to PoseNet. Performance improvement has been seen in [33] by changing the loss function, addition of LSTM after CNN similar to [63] is made to the network in [62, 11] and the network is further improved by making the CNN part Bayesian to estimate uncertainty of localization in [32]. Probabilistic approach has also been explored in [64]. Several approaches[42, 69, 70, 67] have used unsupervised learning approach for depth estimation and VO.

In the realm of inertial odometry (IO) using deep learning, [9] showed improvement over traditional inertial navigation systems like SINS[56] and pedestrian dead reckoning (PDR)[59]. The work presents a deep neural network framework which estimates trajectory by just using raw IMU data with the help of LSTM.

Fusing visual and inertial data also results in increased accuracy of pose estimation. Information fusion in traditional VIO methods is done by two approaches - filtering [41, 46] or non-linear optimization[40, 52]. Multi-state Constraint Kalman Filter[46] is a filtering based method which fuses IMU data with geometric constraints. The optimization based methods include OKVIS[40] and VINS[52] which perform better in accuracy than their filtering based counterparts. Some of the issues which occur in real scenarios that hamper the performance of traditional methods are lighting conditions or occlusions which affect the data captured by the camera, excessive noise and bias in inertial sensor, spatial misalignment, synchronization between the sensors.

In learning based methods for VIO like VINet[12], the image feature extraction is done via another variant of FlowNet[16]. For extracting features from IMU data, an LSTM with the rate at which IMU generates data (which is higher in frequency than visual data) is used. Later, the two features streams are simply concatenated and fed into another LSTM for pose regression. It is the first end-to-end framework for VIO trained in a supervised manner. The work in [10], replaces the naive fusion strategy used in [12] with a more sophisticated one. Two approaches for performing sensor fusion are suggested - Soft (deterministic) fusion and Hard (stochastic) fusion. In soft fusion, the concatenated visual and inertial features are reweighted

by a mask similar to popular attention mechanism[61, 66]. In hard fusion, a binary mask generated by a stochastic function is applied to features which either allows a feature to pass through or blocks it, in contrast to continuous value reweighting of each feature. VIO Learner[58] is an unsupervised scaled trajectory estimation and online error correction work. It uses multiview RGB-D images and inertial data and attempts to minimize the Euclidean loss between the target and reconstructed target image using Jacobians of reprojection errors w.r.t. pixel coordinates. However, the necessity of depth information to recover absolute scale makes it difficult to use as it may not always be available. DeepVIO[26] does VIO in self-supervised manner. It uses optical flow and preintegrated IMU network with a status update module that continuously updates its status similar to traditional tightly coupled VIO approaches. Another unsupervised approach, SelfVIO[1] does monocular VIO and depth reconstruction using adversarial training.

Some works have detailed the importance of uncertainties in the context of deep learning. [34] has outlined various types of uncertainties which are introduced in chapter 2. [27] has incorporated uncertainty in attention mechanism which has been further discussed in section 2.4.4. [53] deals with uncertainty estimation for deep neural networks using Laplace approximation, this has been discussed in the coming sections.

1.3 Contribution and Outline

The objective of the thesis is twofold. Primarily, to estimate the pose of the robotic system (vehicle odometry), an elaborate sensor fusion strategy employing attention is demonstrated. The model is evaluated on challenging datasets using various scenarios of sensor degradation which can be probable in real time, in addition to public datasets viz. KITTI dataset, to verify the notion of complementarity of the employed sensors. A comparison is then made with the existing state-of-the-art works and the effects of these degradations on robustness of all methods are studied extensively. Our benchmark reveals that our approach is able to outperform the state-of-the-art end-to-end VIO methods in terms of accuracy as well as ability to deal with degradations. Secondly, the network is given Bayesian treatment by placing a prior distribution over its weights to estimate model’s uncertainty in predicting the poses. When subjected to data corruptions, the network is able to depict its unreliability by giving more uncertain predictions as compared to normal case.

The thesis is presented in five chapters – Chapter 1 describes the motivation behind the thesis, literature survey and contents. Chapter 2 establishes the theoretical foundation of this thesis by providing a gentle introduction to the building blocks, important methods and concepts for fusion and probabilistic framework for neural networks. The proposed method has been described in detail in Chapter 3 along with an overview of the model architecture and providing Bayesian treatment to it. Chapter 4 describes the current state-of-the-art works, compares the performance of the proposed approach against them and discusses its advantages and important

insights. It also lays basis for the implementation of the Bayesian neural network and showcases important results. Finally, Chapter 5 gives the conclusion and provides a brief outlook for future improvements.

2 Preliminaries

2.1 Basic building blocks

2.1.1 Convolutional Neural Network

The following explanation summarizes the idea of a Convolutional Neural Network (CNN) as described in [31]. A CNN takes an input image, passes it through a number of layers and produces an output which contains important detected features of the image. In a CNN, the first layer is a convolution layer which extracts features from an input image. It performs a mathematical operation of convolution between the image and a filter (kernel). An inner product between them produces a 2D activation map of that filter. The output volume of this layer is controlled by three hyperparameters - depth, stride and padding. To control the number of free parameters, a parameter sharing scheme is used. ReLU is then applied to introduce nonlinearity in the network, it removes negative values from the feature map by setting them to zero. Pooling layer performs down sampling. After a series of convolutional and max pooling layers, comes the fully connected layer where all neurons are connected to all activations in the previous layer. Finally the loss layer specifies the penalty of deviation between the predicted and actual output. Softmax is the preferred function. An example of such a network is shown in Fig. 2.1.

2.1.2 FlowNet

CNNs have been successfully proven to work well with tasks such as classification, segmentation etc. But on temporal domain for e.g. if input is a video sequence,

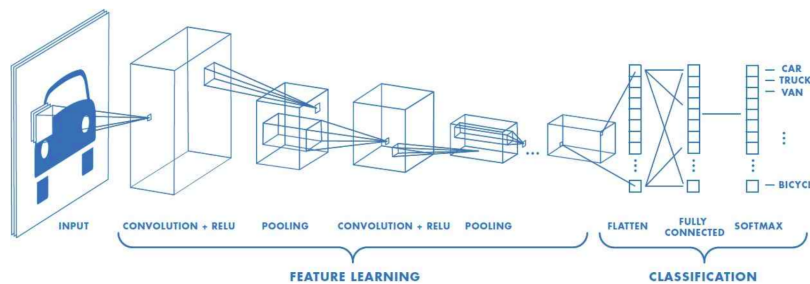


Figure 2.1: Example of a network with many convolutional layers [45], filters are applied to each training image at different resolutions, and the output of each convolved image is used as the input to the next layer

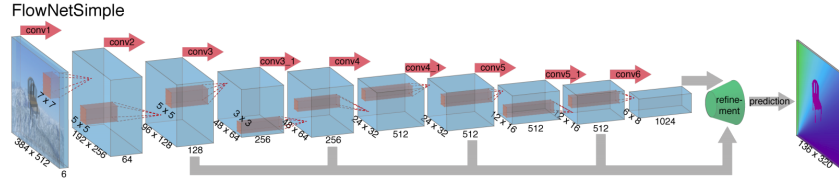


Figure 2.2: FlowNet Simple [16] © 2015 IEEE

their results are less significant. Optical flow is an algorithm that takes into account such temporal information which prioritizes motion as an important characteristic. It is basically a per-pixel localization wherein the pixel brightness motion across the screen over time is estimated[24]. FlowNet[16] is a neural network which estimates optical flow, being trained end-to-end. The prime idea is to learn image feature representation and to match them at different locations in two images. Two identical but separate processing streams for the two images are created and are compared by correlation, combined at a later stage. It is a known fact that interleaving convolutional layers and pooling shrink the feature maps spatially[16]. Hence refinement is used here to provide dense per-pixel prediction. This is done by adding ‘upconvolutional’ layers which consist of unpooling and a convolution. FlowNet Simple is shown in Fig. 2.2.

2.1.3 Recurrent Neural Networks

Classical neural networks have independent inputs and outputs which makes it difficult for them to go back to their previous states. This is particularly important when predicting words in natural language processing or future frames for video processing. This drawback is tackled by recurrent neural networks (RNN) which have loops in them, which thus allow information to be retained. They are designed to learn temporal dependencies.

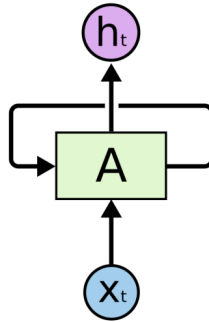


Figure 2.3: Recurrent Neural Network with loop [13]

A simple RNN retains past information in hidden states. The hidden state at the current time step t , h_t is a function of input data at that time instant x_t and the

hidden state from previous time instant h_{t-1} . The memory cell in Fig. 2.3 can be mathematically represented as (as given in [60]),

$$h_t = g(W_{hh}h_{t-1} + W_{hx}x_t) \quad (2.1)$$

Here g is an activation function like sigmoid or hyperbolic tangent. \mathbf{W}_{hh} and \mathbf{W}_{hx} are the weight matrices. The weight matrices determine the importance to be given to the current input and the hidden state from the previous time instant. Their weights are altered via backpropagation through time (BPTT) to reduce loss at each time step.

2.1.4 Long Short-Term Memory

RNNs are not able to learn and handle long term dependencies efficiently. Hence Long Short-Term Memory (LSTM)[28] are used for overcoming this drawback. They tend to remember information for longer periods of time and consist of 4 repeating modules which interact with each other. LSTMs comprise of a cell state and a hidden state. Information can be added to or removed from a cell state by gates. Gates are made of a sigmoid layer and a pointwise multiplication operation. Sigmoid layer decides how much of component to pass through. A value of zero means to let nothing through. For this reason it is called forget gate layer. Next, which new knowledge to be added to the cell state is decided by the hidden state. In the input gate layer, the values to be updated are decided by a sigmoid layer. It also comprises of a \tanh layer which decides on the probable new input values to be added to the state. These two layers are added to create an update to the state. Final stage is the output of LSTM. The cell state is filtered by applying \tanh and then multiplying it by the output of the sigmoid gate, thus deciding to output only the parts that are required.

Like recurrent neural networks, LSTMs can be unfolded at each time step. At time j , for a given input z_j , hidden state h_{j-1} and cell state c_{j-1} from previous time step, the equations for LSTM as given in [63] can be written as

$$\begin{aligned} i_j &= \sigma(\mathbf{W}_{zi}z_j + \mathbf{W}_{hi}h_{j-1} + b_i) \\ f_j &= \sigma(\mathbf{W}_{zf}z_j + \mathbf{W}_{hf}h_{j-1} + b_f) \\ c_j &= f_j \odot c_{j-1} + i_j \odot \tanh(\mathbf{W}_{zg}z_j + \mathbf{W}_{hg}h_{j-1} + b_g) \\ o_j &= \sigma(\mathbf{W}_{zo}z_j + \mathbf{W}_{ho}h_{j-1} + b_o) \\ h_j &= o_j \odot \tanh(c_j) \end{aligned} \quad (2.2)$$

where \tanh is hyperbolic tangent, σ is sigmoid function, weight matrices are denoted by \mathbf{W} , bias terms are represented by b , whereas i_j , f_j , c_j and o_j are input, forget, cell state and output gate at time j respectively. Fig. 2.4 depicts an unrolled LSTM.

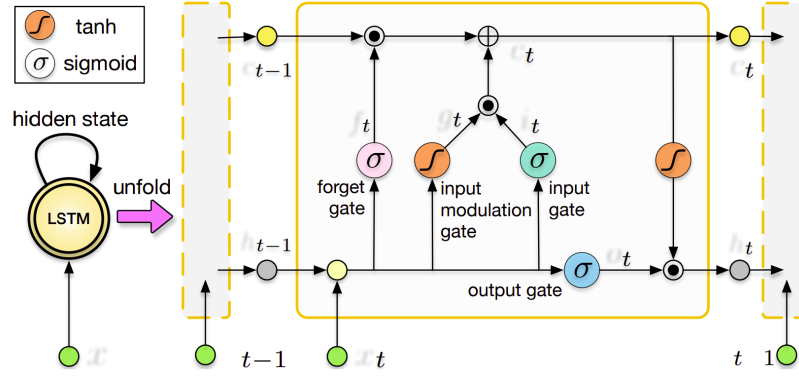


Figure 2.4: Folded and unfolded LSTMs and internal structure of its unit [63]. \odot and \oplus denote element-wise product and addition of two vectors, respectively.
© 2017 IEEE

2.2 Fusion methods

Multimodal fusion can be classified into early, late and intermediate [5, 2, 15] depending upon where the fusion is done.

2.2.1 Early Fusion

In this fusion method, low level input features from different modalities are combined to form a single joint representation before the learning phase. As stated in [5], it just requires a single model and a single learning phase which makes the training pipeline easier when compared to intermediate and late fusion. However this approach suffers from some drawbacks. Features need to be represented in the same format before fusion. Moreover, learning the cross-correlation among various features becomes difficult with an increase in the number of modalities [2].

2.2.2 Late Fusion

This fusion method also known as decision level fusion, involves processing of input features from multiple modalities independently. Later, the unimodal decision values from these independent branches are fused. This approach has significant advantages over early fusion method. A study [15] showed that late fusion approach yielded higher accuracies as compared to early fusion. It offers more flexibility as different models are used for each modality which can model individual modalities in a better way. As decisions have same representation, fusion becomes easier. In addition, it can handle situations where one or more modalities are missing or no parallel data is available as suggested in [5]. However, this approach is unable to use feature level correlation between the modalities. Also, learning process becomes time consuming as different models are used to obtain local decisions [2].

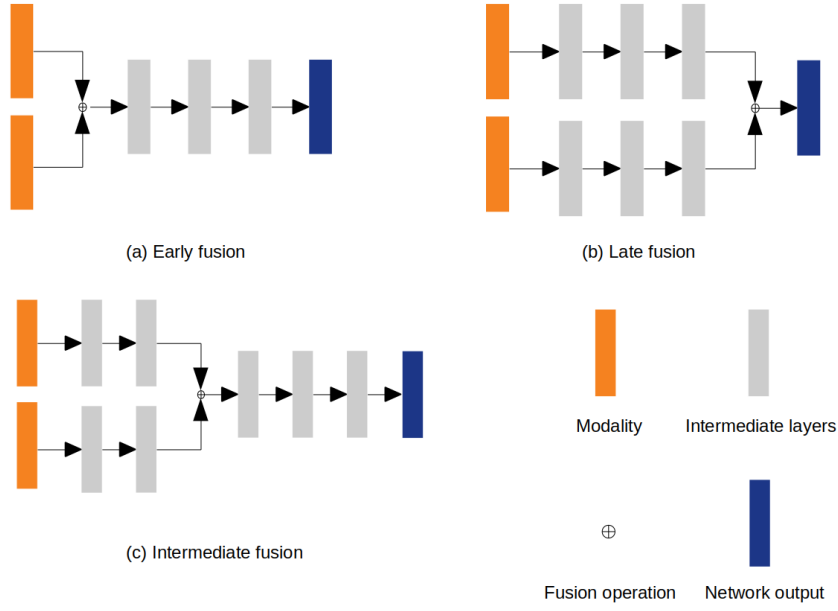


Figure 2.5: An illustration of early fusion, late fusion, and intermediate fusion methods [20]

2.2.3 Intermediate Fusion

Fusion is done at abstract feature levels, for finding a joint data representation. Intermediate fusion attempts to exploit advantages of both of the above mentioned approaches. It also results into significant performance gain in perception tasks [36]. As a consequence we have chosen to follow this approach in our work.

2.3 Self-attention mechanisms

2.3.1 Self-attention

Attention mechanism decides which part of the input to pay attention to at each step of output generation. Generalized attention is calculated by finding a weighted sum of values dependent on the queries and the corresponding keys, for every query Q and a key-value pair (K, V) . The query decides what values to focus on. An alignment model is first computed by performing an operation between query and keys. This operation can be basic dot product, scaled dot-product, multiplicative, additive etc.

Self-attention mechanism, also known as intra-attention as described in [61], relates different parts of the same sequence to compute representation of that sequence. For self attention, $K=V=Q$ of dimension d_k .

$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{QK^\top}{\sqrt{d_k}} \right) V \quad (2.3)$$

Multi-head self-attention[61] does this process of computing attention n times with different, learned weighting matrices for a model of dimension N .

$$\begin{aligned} \text{head}_i &= \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \\ H &= \text{concat}(\text{head}_1, \dots, \text{head}_n) \\ \text{MultiHead}(Q, K, V) &= HW^O \end{aligned} \tag{2.4}$$

Where $W_i^Q \in \mathbb{R}^{N \times d_k}$, $W_i^K \in \mathbb{R}^{N \times d_k}$, $W_i^V \in \mathbb{R}^{N \times d_k}$ and $W^O \in \mathbb{R}^{nd_k \times N}$.

Advantage of using multi-head attention is that it “allows the model to jointly attend to information from different representation subspaces at different positions” [61].

2.3.2 Attention variants

The attention variants have been classified into soft[4, 66] and hard[66].

Soft (Deterministic) attention places the attention weights over all features of the input feature map. Hence features in the focused regions tend to dominate irrelevant ones. The model is differentiable and varies smoothly over its domain. Hence, a softmax function which is differentiable can be used and model can be trained using backpropagation algorithm. But it can be computationally expensive when the input is large. Also, soft attention functions only over discrete spaces.

Hard (Stochastic) attention samples selective features of the feature map and attends to them one at a time. A Monte Carlo based sampling approximation of the gradient is applied. Hard attention is non-deterministic because the focusing region is computed by random sampling, thus making the model non-differentiable. Due to this reason, the model needs to be trained using more complicated techniques like reinforcement learning and variance reduction. An advantage of this type of attention is that the context spaces which are not multinomial can also be attended. This is helpful when the context space is continuous rather than discrete. Hard attention is also computationally less expensive.

2.4 Probabilistic Machine Learning

In Bayesian statistics, probabilities generally quantify uncertainty. The quantities which are not perfectly known are cast as probability distributions which is then followed by their uncertainty quantification. It is assumed that with the help of prior knowledge about the data the probability can be estimated. An update in our beliefs on acquiring new knowledge is done using Bayes’ Theorem [8]:

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)} \tag{2.5}$$

The term on the left hand side represents the posterior over the parameters given data D while the terms on right constitute of likelihood $p(D|\theta)$, prior $p(\theta)$ and evidence $p(D)$. The posterior is proportional to the likelihood, given the constant normalization denominator and a uniform prior [8, 6]. Performing inference (finding suitable parameters) yields the maximum a posteriori (MAP) estimate of the posterior distribution[6]. However, shape of the distribution can not be obtained from these point estimates. A full posterior distribution can incorporate the uncertainty when making predictions considering all possible parameter configurations. This so called posterior predictive distribution is obtained by marginalizing (integrating) over the parameters,

$$p(\tilde{\mathbf{x}}|\tilde{\mathbf{u}}, D) = \int p(\tilde{\mathbf{x}}|\tilde{\mathbf{u}}, \theta)p(\theta|D)d\theta \quad (2.6)$$

Here $\tilde{\mathbf{x}}$ is the predicted target for a new unobserved value $\tilde{\mathbf{u}}$ and θ represents parameters of the model.

2.4.1 Bayesian Neural Networks

Neural networks are prone to overfitting. They are generally not capable of assessing the uncertainty in the training data correctly and this leads to overly confident decisions about the predictions. Bayesian Neural Networks [44, 49] incorporate a measure of uncertainty in the predictions, an aspect which the current neural networks lack.

Bayesian Neural Networks apply prior distribution on network weights θ and give a probability distribution over the weights given the training data, $p(\theta|D)$ rather than having a single most likely value of θ .

2.4.2 Model uncertainty

Model or epistemic uncertainty describes the uncertainty in model parameters that arises due to limited data and knowledge[34]. Neural networks can very well deal with data they have seen before, but are not very good at extrapolation. Epistemic uncertainty is often referred as reducible uncertainty as it is possible to reduce it with more data.

2.4.3 Aleatoric uncertainty

Aleatoric or data uncertainty captures stochasticity in the observations[34]. It can arise due to measurement errors, and is generally termed as irreducible and cannot be done away by collecting more data under the same conditions.

It is further classified into heteroscedastic or data-dependent uncertainty and homoscedastic or task-dependent uncertainty. Heteroscedastic uncertainty depends on the inputs, with some inputs producing more noisy outputs. Homoscedastic uncertainty is not dependent on the input. It is a quantity which is constant for all inputs.

2.4.4 Uncertainty aware attention mechanism

Attention mechanism needs to be incorporated with some sort of uncertainty measure, as neural networks are generally trained in a weakly-supervised manner [27]. An existing mechanism, called Uncertainty-aware Attention (UA)[27], mitigates this limitation by introducing uncertainty to attention mechanism which is input-dependent. A larger variance is associated with inputs that the attention mechanism is uncertain about using variational inference.

In this method, a Gaussian distribution having input-dependent noise is placed on attention weights. Aleatoric uncertainty which varies with inputs can be modelled with input adaptive noise and this results in attenuation of attention strength based on uncertainty. This uncertainty can be further used to make final predictions.

When inputs are often noisy and one-to-one matching with prediction is not possible, model may give incorrect predictions due to the overconfident and inaccurate attentions. This limitation is tackled by this mechanism. If the model is confident about the contribution of a given feature in input, it allocates small variance attention and for uncertain features, it allocates attention with large variance. This suggests that UA can be used for intermediate sensor fusion. However, expected calibration error cannot be calculated for cases where model doesn't give accuracy as an output.

2.4.5 Laplace Approximation

The Laplace Approximation is a deterministic method for approximate inference. It is used for approximating Bayesian parameter estimates and uncertainty in models. The main idea behind it as given by [38, 3] is to find a Gaussian approximation to a posterior by constructing it around the mode of the posterior distribution which is found with the help of numerical optimization. Laplace approximation is a simple two-term second order Taylor expansion of the log posterior around its Maximum A Posteriori (MAP) estimate. For a MAP estimate q_{MAP} , assuming the first order term of the Taylor expansion to be zero, we get:

$$\log p(\mathbf{q}|D) \approx \log p(\mathbf{q}_{MAP}|D) - \frac{1}{2}(\mathbf{q} - \mathbf{q}_{MAP})^\top A(\mathbf{q} - \mathbf{q}_{MAP}) \quad (2.7)$$

Here $A = -\nabla\nabla \log p(\mathbf{q}_{MAP}|D)$ is the average Hessian of the negative log posterior and D is the data. The Hessian, a matrix which consists of second order derivatives, describes the local curvature of the function.

3 Approach

This section focuses on the theory and concepts of the proposed approach in detail. We hereby introduce our approach which makes use of an attention mechanism, viz. Multi-Head Attention along with performing Bayesian inference on layer weights to estimate uncertainty. We select a supervised learning approach for our work due to the availability of accurate ground truth data for training the network. We then see ways for quantification of the uncertainties and their need, because it is very easy to fool a neural network.

3.1 End-to-end monocular VIO architecture

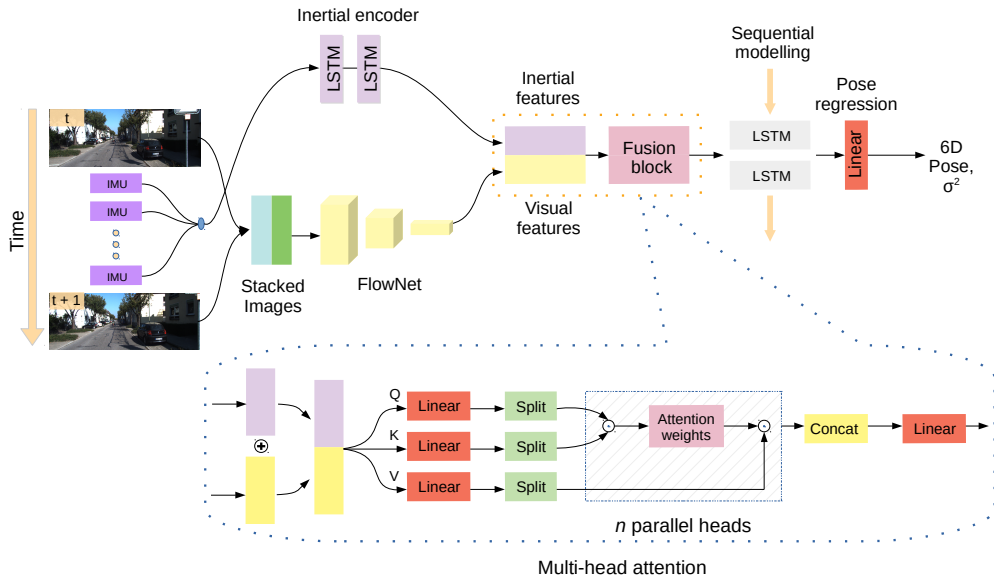


Figure 3.1: An architectural overview of the proposed end-to-end VIO framework with multi-head attention mechanism for sensor fusion. σ^2 denotes the variance, i.e., uncertainty of the poses. Image credit: KITTI dataset. (Adapted from [10])

The modular architecture of the proposed approach for end-to-end VIO is shown

3 Approach

in Fig. 3.1. It consists of feature extractors, fusion mechanism, sequential modelling and pose regression which will now be discussed in detail.

3.1.1 Feature extractors

Visual feature extractor

Two sequential monocular images \mathbf{u}_v are stacked together and fed to the feature extractor g_{vision} . It learns geometric feature representation to be able to generalise well in unfamiliar environments as opposed to learning appearance representation. The employed CNN has its structure adapted from [16] for optical flow estimation. Table 3.1 as given in [63] summarizes its configuration. Firstly, pre-processing of each monocular image is done by normalizing it and then stacking is done. The CNN has total 17 layers, each convolutional layer followed by ReLU activation except for **Conv6**. The CNN learns useful information from the high dimensional image which is evident from the increasing number of channels. This enhances sequential learning of the core LSTM which will be discussed later in section 3.1.3.

Table 3.1: CNN configuration [63] © 2017 IEEE.

Layer	Receptive Field Size	Padding	Stride	Number of Channels
Conv1	7×7	3	2	64
Conv2	5×5	2	2	128
Conv3	5×5	2	2	256
Conv3_1	3×3	1	1	256
Conv4	3×3	1	2	512
Conv4_1	3×3	1	1	512
Conv5	3×3	1	2	512
Conv5_1	3×3	1	1	512
Conv6	3×3	1	2	1024

The visual features \mathbf{b}_v obtained from the final layer of FlowNet can be expressed as:

$$\mathbf{b}_v = g_{\text{vision}}(\mathbf{u}_v) \quad (3.1)$$

Inertial feature extractor

Raw inertial measurements \mathbf{u}_i , i.e. x, y, z components of linear and angular acceleration taken together forming a 6 dimensional vector are passed to inertial encoder

g_{inertial} . N such frames of IMU falling between two consequent image frames are passed. The inertial encoder is composed of a two layer bidirectional IMU-LSTM (see section 2.1.4 for its working) with 15 hidden states each. It operates at a rate at which IMU receives data (which is generally 10 times faster than the frequency of visual data). The LSTM is chosen to be bidirectional as it can preserve and learn information from the past and future of the current point in time, thereby aiding a better understanding of the context. The inertial features \mathbf{b}_i obtained from the LSTMs can be expressed as:

$$\mathbf{b}_i = g_{\text{inertial}}(\mathbf{u}_i) \quad (3.2)$$

3.1.2 Feature fusion mechanism

An overview of different fusion methods was done in section 2.2. Considering the advantages of intermediate sensor fusion, we employ it for our work. This method of fusion combines the high level feature streams generated from raw visual and inertial data. Existing learning based approaches for sensor fusion either simply concatenate the two feature streams into a single feature space or reweight them, thereby leading to suboptimal performance. We therefore introduce a feature fusion mechanism f based on attention[61] to learn a suitable combined feature representation. We will further discuss this mechanism in section 3.2 in detail. The attention guided function f now fuses the visual \mathbf{b}_v and inertial \mathbf{b}_i feature vectors to produce a combined feature representation \mathbf{y} which will be then passed to pose regression module:

$$\mathbf{y} = f(\mathbf{b}_v, \mathbf{b}_i) \quad (3.3)$$

3.1.3 Sequential modelling and pose regression

Accurate pose estimation is a direct result of modelling sequential dependence, which is one of the key aspects of ego motion estimation. This temporal modelling is done by a recurrent neural network, viz. the core LSTM. The resultant feature representation from the feature fusion block \mathbf{y}_t at time t is fed to the core LSTM along with its hidden states from the previous time step \mathbf{h}_{t-1} . It has two layers of LSTM stacked together in order to learn complex model dynamics (motion model) and to derive connections between sequential features. Each LSTM layer has 1000 hidden states, and the output of the first layer is input to the next. The fully connected (FC) layer does the pose regression and the output \mathbf{x}_t is a 6D camera pose - composed of 3D Euler angles and 3D translation.

$$\mathbf{x}_t = f_{\text{lstm}}(\mathbf{y}_t, \mathbf{h}_{t-1}) \quad (3.4)$$

3.1.4 Loss function

We use the same cost function defined in [63]. We wish to find the conditional probability of poses \mathbf{X}_k given sequential sensor data \mathbf{U}_k till time instance k given by,

$$p(\mathbf{X}_k|\mathbf{U}_k) = p(\mathbf{x}_1, \dots, \mathbf{x}_k|\mathbf{u}_1, \dots, \mathbf{u}_k) \quad (3.5)$$

The optimal parameters λ^* are found by maximizing eq. 3.5,

$$\lambda^* = \underset{\lambda}{\operatorname{argmax}} p(\mathbf{X}_k|\mathbf{U}_k; \lambda) \quad (3.6)$$

These hyperparameters can be learned by minimizing the loss function,

$$\lambda^* = \underset{\lambda}{\operatorname{argmin}} \ell(f_\lambda(\mathbf{U}_k), \mathbf{X}_k) \quad (3.7)$$

where f_λ is a function that maps sensor data to poses.

The loss function constitutes of the summation of Euclidean distance (mean squared error (MSE)) between predicted poses $(\hat{\mathbf{z}}_t, \hat{\psi}_t)$ and ground truth (\mathbf{z}_t, ψ_t) where \mathbf{z}_t denotes position and ψ_t the orientation at time t ,

$$\ell = \frac{1}{M} \sum_{i=1}^M \sum_{t=1}^k \|\hat{\mathbf{z}}_t - \mathbf{z}_t\|_2^2 + \beta \|\hat{\psi}_t - \psi_t\|_2^2 \quad (3.8)$$

where β (chosen to be 1000) is a scale factor to balance the position and orientation elements with total M samples. The reason for choosing orientation ψ as Euler angle representation is because quaternions impede the underlying optimization as they have an additional unit constraint and this results in some orientation degradation [63, 68].

3.2 Multi-Head Self-Attention

Visual and inertial sensors have a unique complementarity. For ego motion estimation tasks, the exteroceptive monocular visual sensor measures geometry and appearance of the environment, but suffers from scale ambiguity, whereas egocentric inertial sensor makes the metric scale observable and provides with reliable motion estimates even in the case of loss of visual tracking [21]. However, both sensors have their own shortcomings. Motion blur, poor illumination conditions and lack of features can cause erroneous data associations in visual sensor. Moreover, inertial sensors are plagued with noise and bias. In case of such sensor degradation scenarios, simply considering all the features for fusion may prove to be catastrophic and lead to erroneous estimates. Making the network “attentive” to focus on important features alleviates this difficulty.

In our work, we employ the multi-head self-attention mechanism for feature selection as already introduced in section 2.3.1. The visual and inertial feature representations obtained from the respective encoders are combined together to form a concatenated feature vector. This combined feature vector is then fed into the multi-head attention module. The use of self attention stems from the fact that for a given context it is capable of modelling long range interactions. As the combined feature representation is treated as query, key and value, the self-attention mechanism compares the query with the key by dot product operation[61]. The choice of dot product as a scoring function is because it is computationally faster and efficient. It is then scaled by $1/\sqrt{d_k}$, where d_k is the dimension of the representation. Softmax function is then applied to it to obtain attention weights (probabilities), which indicate importance given to corresponding inputs. With the increase in input dimension d_k , the dot product becomes large and as a result the softmax function may have extremely small gradients[61]. This is avoided by the scaling operation. In order to learn distinct representations of the input, many such attention *heads* are employed, as seen in equation 2.4. The output from each head h is then concatenated to produce final attention outputs for a given combined feature representation.

3.3 Uncertainty estimation

In section 2.4.5, we saw an introduction to Laplace’s method for estimating uncertainty in neural networks, which has been further extended to work with deep neural networks in [53].

We model uncertainty by placing a prior distribution on weights of the model, approximate the intractable posterior and analyse the variance in the weights given different data. Rewriting equation 2.7 in terms of MAP estimate of the posterior θ_{MAP} :

$$\log p(\theta|D) \approx \log p(\theta_{MAP}|D) - \frac{1}{2}(\theta - \theta_{MAP})^\top A(\theta - \theta_{MAP}) \quad (3.9)$$

This approximation is only well defined when the average Hessian A is positive semidefinite, meaning θ_{MAP} must be a local maximum[8, 53]. Exponential of the above equation gives:

$$p(\theta|D) \approx p(\theta_{MAP}|D) \exp \left(-\frac{1}{2}(\theta - \theta_{MAP})^\top A(\theta - \theta_{MAP}) \right) \quad (3.10)$$

which depicts the probability density function of Gaussian distribution. The Gaussian approximation of the posterior over weights is given by,

$$\theta \sim \mathcal{N}(\theta_{MAP}, A^{-1}) \quad (3.11)$$

We perform Bayesian inference to approximate the posterior mean when dealing with

3 Approach

unseen data by taking average of $T = 30$ Monte Carlo samples θ_* [53], which is the Monte Carlo approximation of the integral,

$$\begin{aligned} p(\tilde{\mathbf{x}}|\tilde{\mathbf{u}}, D) &= \int p(\tilde{\mathbf{x}}|\tilde{\mathbf{u}}, \theta) p(\theta|D) d\theta \\ &\approx \frac{1}{T} \sum_{i=1}^T p(\tilde{\mathbf{x}}|\tilde{\mathbf{u}}, \theta_*), \quad \theta_* \sim p(\theta|D) \end{aligned} \quad (3.12)$$

3.3.1 Fisher information matrix

In practice, the number of weights in a neural network are of the order of tens of thousands. It becomes infeasible to calculate the Hessian w.r.t. these weights or even to invert it. An easily computable approximation of the Hessian is the diagonal of the Fisher information matrix (Fisher). As $p(\mathbf{x}|\mathbf{u}, \theta)$ is generally not known, empirical Fisher F is used in place of true one, which is given by the expectation of squared gradients as[53],

$$F = \mathbb{E} \left[\nabla_{\theta} \log p(\mathbf{x}|\mathbf{u}) \nabla_{\theta} \log p(\mathbf{x}|\mathbf{u})^{\top} \right] = \mathbb{E} \left[(\nabla_{\theta} \log p(\mathbf{x}|\mathbf{u}))^2 \right] \quad (3.13)$$

$$A \approx \text{diag}(F) \quad (3.14)$$

where $\text{diag}(F)$ returns a diagonal matrix of the full Fisher matrix, because for diagonal approximation, the covariance matrix reduces to its diagonal.

For every layer l of the network, this is equivalent to sampling weights from normal distribution with diagonal covariance,

$$\theta_l \sim \mathcal{N}(\theta_{MAP,l}, \text{diag}(F)^{-1}) \quad (3.15)$$

3.3.2 Parameters of Laplace method

[53] pointed out that it is beneficial to regularize curvature matrices like Fisher for these reasons:

1. Laplace's method makes some approximations such as - independence between the layers ignoring covariance between them, and approximation of expectation in order for the posterior to become tractable. This may lead to an overestimation of variance in certain directions.
2. If some weights exhibit high covariance, the Laplace approximation might place a considerable probability mass in low probability regions of the true posterior.

A simple regularization scheme has been introduced in [53] which has also been used in this thesis. The Fisher for each layer l , F_l is scaled by N which is the size of the dataset and incorporates a Gaussian prior on the weights τ [8],

$$NF_l + \tau I \quad (3.16)$$

Here, identity matrix is multiplied to the prior.

3.3.3 Sampling weights

We sample weights θ from a normal distribution around the mean, represented by trained weights θ_{MAP} and covariance matrix $\text{diag}(F)^{-1}$ depicting the inverse curvature factors as,

$$\theta = \theta_{MAP} + \mathbf{V} \mathbf{s} \quad (3.17)$$

where samples drawn from a normal distribution with mean zero and variance one, i.e. standard normal distribution are represented by \mathbf{s} and $\mathbf{V} \mathbf{V}^\top$ is the Cholesky decomposition of $\text{diag}(F)^{-1} \in \mathbb{R}^{m \times m}$ with $\text{Cholesky}(\mathbf{V}) = \sqrt{\mathbf{V}}$ for a diagonal matrix[53].

Laplace's method can be summarized by a series of following steps:

- 1) Select a trained network model.
- 2) Compute diagonal Fisher (3.3.1).
- 3) Sample weights (3.3.3).
- 4) For each sampled weight configuration:
 - a. Replace old weights with the new configuration.
 - b. Calculate output.
- 5) Evaluate uncertainty on collective output (3.4).

3.4 Uncertainty quantification

In order to measure the quality of uncertainty estimates, we define metrics such as mean squared error and variance.

3.4.1 Mean Squared Error

Mean squared error (MSE) is given by:

$$L(\theta) \triangleq \frac{1}{M} E(\theta) \quad (3.18)$$

$$E(\theta) = \sum_{i=1}^M (\mathbf{x} - \hat{\mathbf{x}})^2 \quad (3.19)$$

3 Approach

Equation 3.19 can be interpreted probabilistically and can be expressed as the negative log of multivariate normal distribution \mathcal{N} with precision β [29].

$$E(\theta) = -\ln \exp \left(\frac{-\sum_{i=1}^M (\mathbf{x} - \hat{\mathbf{x}})^2}{2\sigma^2} \right) \propto -\ln \mathcal{N}(\mathbf{x}|\hat{\mathbf{x}}, \beta^{-1}I) \quad (3.20)$$

The above quantity is also called negative log likelihood (NLL). Since this is same as MSE defined in equation 3.8, we use it as a metric for uncertainty estimation as well.

3.4.2 Variance

The variance of some function $g(x)$, denoted by σ^2 is given by [8].

$$\text{var}[g] \triangleq \mathbb{E} \left[(g(x) - \mathbb{E}[g(x)])^2 \right] \quad (3.21)$$

where $\mathbb{E}[g(x)]$ is the mean of the function.

3.5 Prediction reliability of neural networks

It is very easy to fool neural networks by subjecting them to perturbed inputs, which misleads them to give wrong estimates[25]. In such cases, it is often useful to get an indication from the network that it is uncertain about such inputs instead of getting an overconfident wrong prediction. Such perturbations can easily occur in real-life scenarios due to various reasons. We will now see these reasons and consequently the generation of such instances.

3.5.1 Degradation scenarios

We prepare two groups of datasets by introducing various types of sensor degradation as done in [10]. They are as under:

Visual degradation

1. Part occlusion: We cutout a part of dimension 200×200 pixels from an image at a random location by overlaying a mask of the same size on it. Such situation can occur when camera view is obstructed by objects very close to it or due to dust[65].
2. Noise: We add salt and pepper noise along with Gaussian blur to the images. This can occur due to substantial horizontal motion of the camera, changing light conditions or due to defocusing[14].
3. Missing frames: Some images are removed at random. This happens when the sensor temporarily gets disconnected or while passing through low-lit area like a tunnel.

Inertial degradation

1. Noise: IMU has inherent errors due to biases in accelerometers and gyroscopes and gyro drift. In addition to the already existing noise we add white noise to accelerometer and bias to gyroscope. This can happen due to temperature variations giving rise to white noise and random walk[17].
2. Missing frames: Random removal of IMU frames between two image frames is done. This is plausible due to packet loss from bus or sensor instability.

The sensor corruptions tell a lot about network robustness. Robustness of the network to such degradations provides an insight into the underlying sensor fusion mechanism. It points towards the resilience of the network to sensor corruption and consequent failure. It highlights the complementary nature of the sensor modalities in face of adversity. It also gives an insight into how different fusion strategies give relative importance to visual and inertial features.

4 Experiments, Results and Discussions

We begin our analysis by firstly defining relevant baselines as described in Section 1.2. We then present a comparison of their performance with our approach. Lastly, we present the empirical analysis of the incorporated Bayesian framework for localization uncertainty.

The baselines and proposed method are all implemented in Python and PyTorch[51].

4.1 Baseline setup and implementation

In this section we first introduce the considered relevant baselines and dataset.

4.1.1 Baselines

A list of baselines used for this work along with their characteristics are described below. These works being state-of-the-art currently in the realm of supervised learning based VO and VIO, are chosen as baselines to be compared against in the thesis, because our work also falls under the same category.

1. The first baseline we consider is DeepVO[63]. It consists of a CNN adapted from [16] for optical flow estimation to which two consecutive stacked monocular images are fed. The subsequent RNN which does temporal modelling has two layers of LSTM stacked together with 1000 hidden states each, and the output of the first layer is input to the next. Pose regression is done by an FC layer to give 6 dimensional pose of the camera.
2. The second baseline is a visual-inertial odometry framework VINet[12]. Similar to [63], it also uses CNN for generating optical flow. $N \times 6$ dimensional IMU data consisting of linear and angular acceleration is passed to IMU-LSTM which processes data at IMU rate. The feature representations from CNN and IMU-LSTM are then concatenated naively before being fed to the core LSTM. A final $SE(3)$ concatenation layer then gives the 7 dimensional $SE(3)$ pose - 3D translation and 4D quaternion as output. For this thesis, the $SE(3)$ composition layer has been removed as no other recent works make use of it due to the fact that fully connected layer can do the pose regression well. Our implementation of VINet is better in terms of accuracy and robustness than what has been reported in the original work.

3. The final baseline is selective sensor fusion approach[10], in which the visual and inertial feature extraction pipeline remains same as described in the works above. We select the soft (deterministic) fusion approach for setting up a baseline as it is similar to attention mechanism that has been used in our work. The soft fusion function does the re-weighting of each feature and is differentiable. Firstly, a mask \mathbf{m} is constructed from concatenation of visual features \mathbf{f}_V and inertial features \mathbf{f}_I

$$\mathbf{m} = \text{Sigmoid}(\mathbf{W}_m[\mathbf{f}_V; \mathbf{f}_I]) \quad (4.1)$$

where \mathbf{W}_m are the weights. The features are reweighted between $[0, 1]$ by the sigmoid function. This mask is then applied to the concatenated features by element-wise multiplication to get new reweighted features according to their relative importance

$$\mathbf{W}_{soft} = \mathbf{m} \odot [\mathbf{f}_V; \mathbf{f}_I] \quad (4.2)$$

where the resultant weight matrix \mathbf{W}_{soft} is then fed to core LSTM - FC layers for sequential modelling and pose regression. An overview of the approach can be seen in Fig. 4.1. We will henceforth refer this work as VIO Soft.

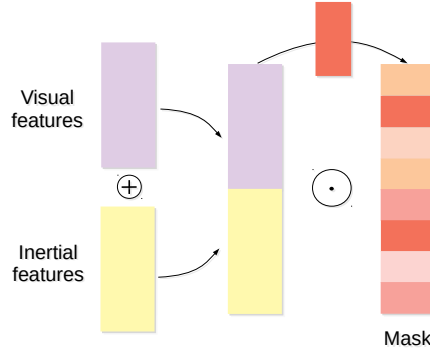


Figure 4.1: Soft fusion [10]

4.1.2 Dataset

The KITTI odometry benchmark[22] for odometry evaluations consists of 22 sequences amounting to a total of 39.2 km length. Images are acquired at the rate of 10 Hz and saved in png format. OXTS RT 3003 is a GPS/IMU localization system

that captures IMU data at 100 Hz (found in raw KITTI dataset) and also determines the ground truth data for this dataset. The ground truth is also acquired at 10 Hz. Sequences 00-10 have corresponding ground truth data associated with them for training whereas sequences 11-22 are without corresponding ground truth for evaluation purposes.

4.1.3 Evaluation metric

We analyse our methods with the help of metrics (reproduced from [23]) for KITTI dataset. As provided in [58], we evaluate translational RMSE in percentage $t_{rel}(\%)$ and rotational RMSE $r_{rel}(^\circ)$ per 100 m, the average value of which is computed on subsequences of length 100m - 800m :

$$\begin{aligned} E_{rot}(\mathcal{F}) &= \frac{1}{|\mathcal{F}|} \sum_{(i,j) \in \mathcal{F}} \|(\hat{\mathbf{r}}_i \ominus \hat{\mathbf{r}}_j) \ominus (\mathbf{r}_i \ominus \mathbf{r}_j)\|_2 \\ E_{trans}(\mathcal{F}) &= \frac{1}{|\mathcal{F}|} \sum_{(i,j) \in \mathcal{F}} \|(\hat{\mathbf{p}}_i \ominus \hat{\mathbf{p}}_j) \ominus (\mathbf{p}_i \ominus \mathbf{p}_j)\|_2 \end{aligned} \quad (4.3)$$

where \mathcal{F} is a set of frames, $[\mathbf{p}, \mathbf{r}] \in SE(3)$ and $[\hat{\mathbf{p}}, \hat{\mathbf{r}}] \in SE(3)$ are true and estimated pose values lying in Lie group $SE(3)$ respectively, and \ominus is the inverse compositional operator.

4.2 Validation

4.2.1 Training and testing

We use the sequences 00, 01, 02, 05, 08, 09 for training because they are comparatively longer and sequences 04, 06, 07, 10 for testing. Sequence 03 is omitted due to missing raw data file. In order to generate more training data, the sequences are segmented into trajectories of varying lengths which in turn avoids overfitting by core LSTM. Here the varying sequence lengths is a hyperparameter.

The network is trained on an NVIDIA GTX 1080 GPU. For DeepVO, Adagrad optimizer with a learning rate of $5e^{-4}$ is used, the network is trained for 250 epochs. Transfer learning is done using pre-trained FlowNet weights[16] to reduce time required for training. VINet is trained using Adam optimizer with learning rate of $1e^{-4}$. VIO Soft is trained using Adam optimizer with a learning rate of $1e^{-5}$. For our method (we refer to it as MHA (Multi-Head Attention) henceforth), transfer learning is done from VINet model for both feature extractors, again using Adam optimizer with learning rate of $5e^{-5}$. Regularization techniques like batch-normalization and dropout have been used.

The performance of our approach and all the three baselines on test sequences 06 and 10 w.r.t. ground truth has been shown in Fig. 4.2. Table 4.1 compares them according to the metrics described in section 4.1.3.

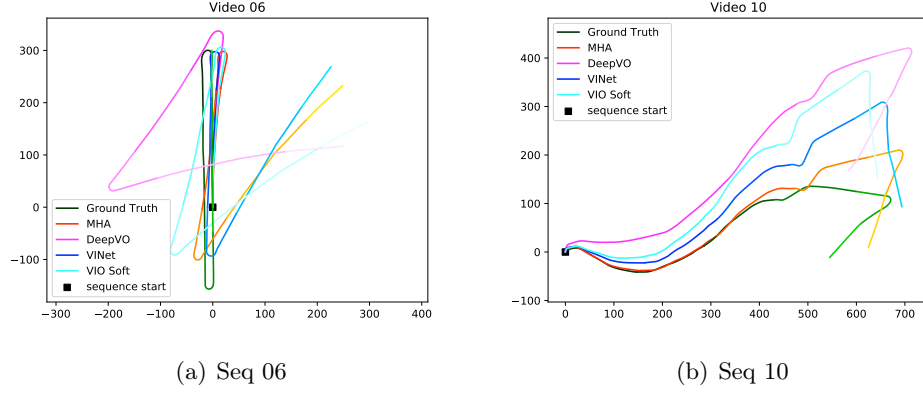


Figure 4.2: Predicted trajectories of the KITTI dataset

Table 4.1: Comparison metric for nominal case for sequences 04, 06, 09, 10

Seq	DeepVO		VINet		VIO Soft		MHA	
	$t_{rel}(\%)$	$r_{rel}(\circ)$	$t_{rel}(\%)$	$r_{rel}(\circ)$	$t_{rel}(\%)$	$r_{rel}(\circ)$	$t_{rel}(\%)$	$r_{rel}(\circ)$
04	6.878	2.484	11.300	2.637	10.841	2.421	8.902	1.586
06	26.520	8.907	15.335	5.256	17.753	6.912	15.592	5.564
09	18.639	4.686	6.954	1.794	6.982	2.264	8.387	2.801
10	23.172	4.332	22.341	8.988	19.030	7.746	12.821	5.809
Mean	18.077	5.882	13.982	4.668	13.651	4.835	11.425	3.94

4.2.2 Empirical analysis

In order to make extensive evaluation of our approach against the our baselines, we test them with various types of sensor degradation introduced in section 3.5.1 and study their effects.

All types of degradation have been applied to 100% of the test dataset. Fig. 4.3 shows the performance for occlusion degradation dataset. Table 4.2 shows comparison metrics for the same. Fig. 4.4 and Table 4.3 do it for IMU degradation scenario. Fig. 4.5 and Table 4.4 depict the performances for all vision degradation scenario(occlusion, noise, 50% missing data) respectively. In the end Fig. 4.6 and Table 4.5 do it for all sensor degradation scenario.

Looking at Fig. 4.2 and Table 4.1, it can be seen that VINet shows improvement over DeepVO due to the addition of IMU, meaning the model gets more data to learn information from and hence the final estimated trajectories are more accurate. However, by replacing the naive concatenation with soft fusion layer with importance mask, the task of training the network became hard due to the addition of soft fusion

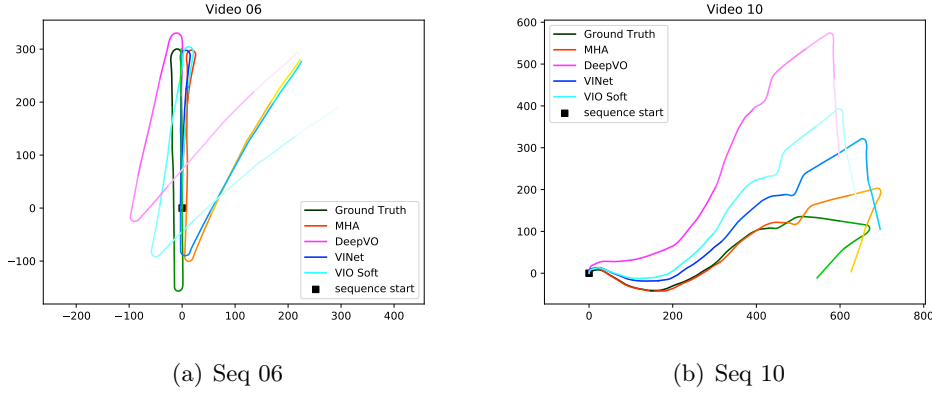


Figure 4.3: Predicted trajectories of the KITTI dataset for occlusion degradation

Table 4.2: Comparison metric for occlusion vision degradation case

Seq	DeepVO		VINet		VIO Soft		MHA	
	$t_{rel}(\%)$	$r_{rel}(^\circ)$	$t_{rel}(\%)$	$r_{rel}(^\circ)$	$t_{rel}(\%)$	$r_{rel}(^\circ)$	$t_{rel}(\%)$	$r_{rel}(^\circ)$
04	10.419	3.855	11.772	2.710	11.938	2.360	9.898	1.787
06	23.026	6.793	16.333	5.581	18.039	6.815	14.919	5.270
09	16.786	5.117	7.368	1.965	6.631	2.020	9.816	3.180
10	26.168	5.547	22.830	9.267	19.142	8.205	13.515	6.178
Mean	19.099	5.328	14.575	4.880	13.937	4.85	12.037	4.103

layer. Due to this, pretrained weights of CNN and IMU-LSTM from VINet model have been used for VIO Soft. As expected, the performance of VIO Soft is nearly similar to VINet. Using the multi-head self-attention layer for feature fusion with 2 heads helps the model to focus on different parts of the feature space. Since it is more representative, the model learns best suitable feature combination, which is advantageous, because equal reliability can't be guaranteed for all features (as is the case with VINet). Hence our method performs significantly better.

For occlusion vision degradation case, it can be seen in Fig. 4.3 that performance of DeepVO degrades with degraded data although CNNs are robust to illumination and occlusions. Due to presence of IMU data, other baselines and our method perform quite well in this scenario.

From Table 4.3, it can be seen that as the sensor fusion strategy becomes elaborate for each approach, the robustness to IMU degradation increases. For low to moderate intensity of data degradation, the performance of all methods remains similar to nominal case, suggesting visual features are more dominant than inertial ones.

For all vision degradation case, as seen in Fig. 4.5 performance of DeepVO de-

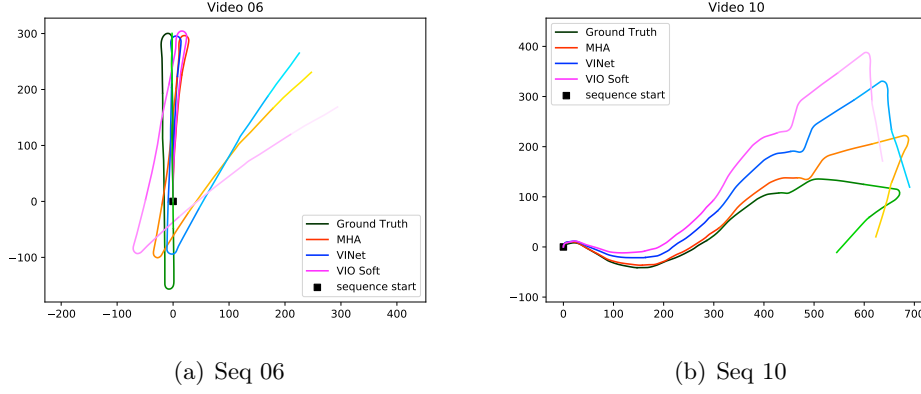


Figure 4.4: Predicted trajectories of the KITTI dataset for IMU degradation

Table 4.3: Comparison metric for IMU degradation case

Seq	DeepVO		VINet		VIO Soft		MHA	
	$t_{rel}(\%)$	$r_{rel}(^\circ)$	$t_{rel}(\%)$	$r_{rel}(^\circ)$	$t_{rel}(\%)$	$r_{rel}(^\circ)$	$t_{rel}(\%)$	$r_{rel}(^\circ)$
04	N/A	N/A	12.564	3.022	10.5634	2.955	9.303	1.607
06	N/A	N/A	16.712	5.319	18.873	7.166	17.046	5.826
09	N/A	N/A	10.476	3.504	7.719	2.336	7.407	2.139
10	N/A	N/A	25.489	9.284	19.079	7.337	16.188	6.157
Mean	-	-	16.310	5.282	14.058	4.948	12.486	3.932

grades immensely and tracking is lost completely. This is expected because the dataset is severely degraded. We can clearly see the advantage of adding another sensor as other baselines and our method appear to be robust to it. Interesting to observe is, that when vision degrades, the translational error increases for all the approaches, but the rotational error remains comparable to nominal case. This means that the visual features contribute to determining translation and inertial contribute to rotation. Inertial features become more reliable in case of strong visual degradation.

Lastly, the evaluation has been made for the case where the dataset is corrupted using all five degradations. Table 4.5 shows the robustness of all approaches even in the presence of strong corruption in all sensor data. Looking at all the degradation scenarios, it can be seen that our method significantly outperforms the baselines.

From the last two degradation scenarios - all vision degradation and all sensor degradation, an important observation can be made in context of correlation between features and dynamics of environment. The rotational error in the latter case increases due to addition of inertial corruption as compared to former case, while the

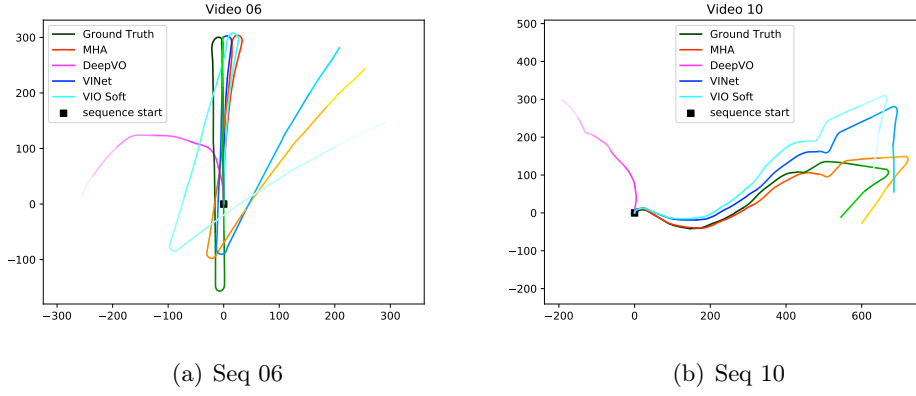


Figure 4.5: Predicted trajectories of the KITTI dataset for all vision degradation

Table 4.4: Comparison metric for all vision degradation case

Seq	DeepVO		VINet		VIO Soft		MHA	
	$t_{rel}(\%)$	$r_{rel}(^\circ)$	$t_{rel}(\%)$	$r_{rel}(^\circ)$	$t_{rel}(\%)$	$r_{rel}(^\circ)$	$t_{rel}(\%)$	$r_{rel}(^\circ)$
04	F	F	12.293	2.838	11.827	2.430	9.615	1.908
06	A	A	14.801	5.089	19.019	6.789	16.755	5.678
09	I	I	10.113	3.395	7.448	2.204	10.228	2.840
10	L	L	24.671	8.558	20.069	6.925	13.864	5.532
Mean	-	-	15.469	4.97	14.590	4.587	12.615	3.989

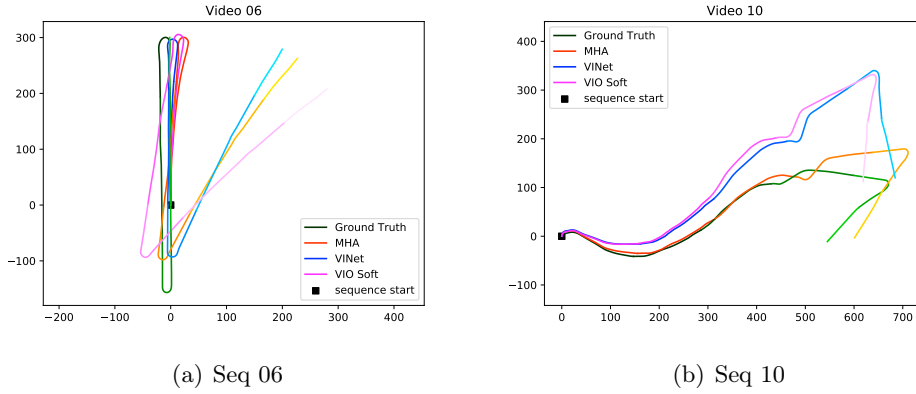


Figure 4.6: Predicted trajectories of the KITTI dataset for all sensor degradation

translational error remains more or less the same. This suggests that inertial data

Table 4.5: Comparison metric for all sensor degradation case

Seq	DeepVO		VINet		VIO Soft		MHA	
	$t_{rel}(\%)$	$r_{rel}(^\circ)$	$t_{rel}(\%)$	$r_{rel}(^\circ)$	$t_{rel}(\%)$	$r_{rel}(^\circ)$	$t_{rel}(\%)$	$r_{rel}(^\circ)$
04	N/A	N/A	13.593	3.090	12.985	2.613	10.679	2.033
06	N/A	N/A	15.001	4.981	16.948	5.915	15.385	5.178
09	N/A	N/A	10.393	3.508	9.344	2.718	9.934	2.667
10	N/A	N/A	24.494	9.008	22.652	7.288	13.882	5.305
Mean	-	-	15.87	5.146	15.482	4.633	12.470	3.795

contributes more to rotation while visual data contributes to translation.

The performances relating to noise+blur and missing data degradation cases have not been included here. No obvious difference was observed as compared to nominal case since FlowNet is robust to deal with such small visual degradation.

To summarize, it can be seen that exploiting the combination of sensors proves to be beneficial for odometry in comparison to use of single sensor (DeepVO). Among all three sensor fusion approaches, our approach being more expressive than naive VINet and VIO Soft, performs better than them in terms of accuracy (lower translational and orientation error). Our method appears to be more robust to sensor corruptions and tends to diverge less as compared to the other two approaches.

4.3 Sensor fusion representation in VIO Soft

In order to have an insight into how sensor fusion is taking place inside the network for VIO Soft, we tried examining various cases for degradation. Fig. 4.7 shows the weights of the mask for the image frame above. Looking at the feature masks, it can be seen that visual features are given less importance for occlusion in comparison to nominal case, and inertial features are given comparatively more importance. It was also observed that internally, weights for IMU features are higher in magnitude than visual features.

However, the importance mask for IMU degradation case is exactly similar to nominal case due to the fact that IMU features themselves have smaller influence. Moreover, it has been observed that during special cases like turning motion, IMU features don't get more importance as mentioned in [10]. Also the trend seen in Fig. 4.7 is not evident at all times. From the metric comparison it can be seen that the reweighting scheme of VIO Soft tends to perform better than naive feature concatenation of VINet, however how strongly these importance masks influence the feature selection is still not clear.

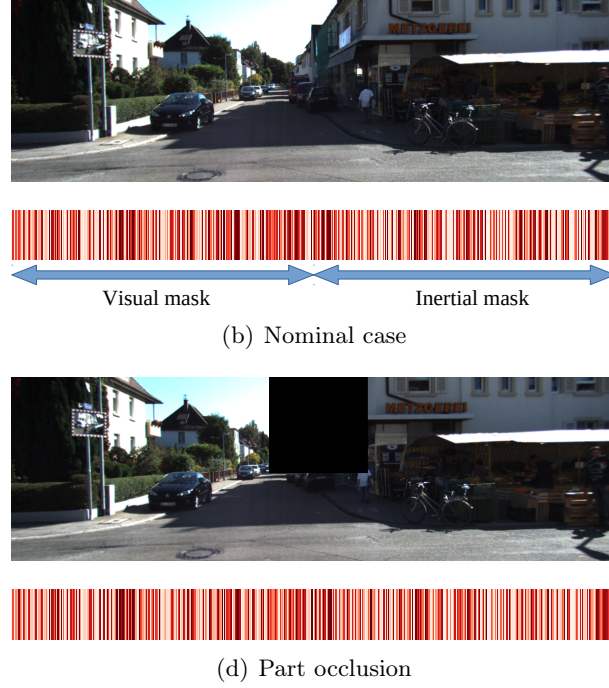


Figure 4.7: Soft fusion mask weights for nominal and occlusion degradation case

4.4 Analytical outlook on network expressivity

The baselines VINet and VIO Soft either use simple concatenation ($\text{concat}(\mathbf{u}, \mathbf{v})$) or a linear layer $f(\mathbf{u}, \mathbf{v}) = \mathbf{W}\text{concat}(\mathbf{u}, \mathbf{v}) + \mathbf{a}$, to fuse information from two different input modalities \mathbf{u} and \mathbf{v} . In contrast to this, our approach is based on considering the effects of input streams on one another, namely *multiplicative interactions* (MI). They can be expressed by the following function [30],

$$f(\mathbf{u}, \mathbf{v}) = \mathbf{v}^\top \mathcal{W} \mathbf{u} + \mathbf{v}^\top \mathbf{W}_v + \mathbf{W}_u \mathbf{u} + \mathbf{a} \quad (4.4)$$

wherein the weight matrices \mathbf{W}_u and \mathbf{W}_v , vector \mathbf{a} and 3D weight tensor \mathcal{W} are the learned parameters. A major difference to additive interaction is the product term $\mathbf{v}^\top \mathcal{W} \mathbf{u}$.

[30] shows that the expressivity of the network using MI is improved as compared to using linear layers. MI tend to expand the hypotheses space – the set of all possible functional mappings that give correct outputs, of the latter case; thereby enabling them to inculcate better contextual information of the fusion task. This flexible form helps in learning the correct inductive bias, i.e. preferential selection of some hypotheses over the rest. Hence, we are able to observe a performance gain using our approach.

4.5 Bayesian framework

We now evaluate our approach for estimating uncertainty in its predictions. For this, the pretrained network is converted into a Bayesian Neural Network using the Laplace approximation strategy without any additional changes to it. The choice of this strategy stems from the fact that it is a practical method not involving the need to redesign the model. The trained model is directly used to perform inference.

4.5.1 Curvature approximation and sampling

From the trained network, we obtain the MAP estimate of the weights of the network. We now compute the curvature approximation - the diagonal Fisher using the entire training set. To do this, the poses are sampled from the output distribution of the trained model. The reason to use samples from output distribution is because we want the actual Fisher information matrix to be identical to the Hessian (in the limit of infinitely many samples). Later, an update of the curvature estimator is done. Once the diagonal Fisher information matrix is computed, weight configurations are sampled from the distribution as described in Section 3.3.3.

4.5.2 Hyperparameters of the BNN

Finding suitable hyperparameters τ and N (eq. 3.16) is a crucial step towards obtaining quality estimates of uncertainty. This explanation summarizes the idea from [29]. Their values relate to damping of the curvature, wherein higher values correspond to higher damping, robbing the BNN of its ability to produce uncertainty estimates (making it behave like the deterministic neural network). In order to capture uncertainty, we need to find parameters that provide optimal damping. However, due to limited knowledge about their plausible values, it is a challenging task to arrive at the optimal values. Moreover, the comprehensional insufficiency in their representational significance and their individual as well as joint effect adds to the difficulty. We therefore select a search space spanned in powers of 10 (log-space) for a wider coverage. A random or a grid search could be performed to find suitable parameters, however these approaches are extremely time consuming. This is because they require as much forward passes as the weight samples using the whole training set per evaluation. Due to time constraints, workable values were found through manual search in this work.

4.5.3 Results for uncertainty estimates

We perform approximate Bayesian inference by evaluating the posterior predictive distribution (eq. 3.12) from sampled weight configurations obtained in section 4.5.1. We then take the mean of the predictions and the uncertainty in those pose predictions is given by their variance.

The grey shaded region around the trajectory in the plots below represents uncertainty in mean predicted poses for $\log\tau = 6.795$ and $\log N = 5.568$. From Fig.

4.8b and Fig. 4.9b it can be seen that increased variance expresses the models' uncertainty about the point.

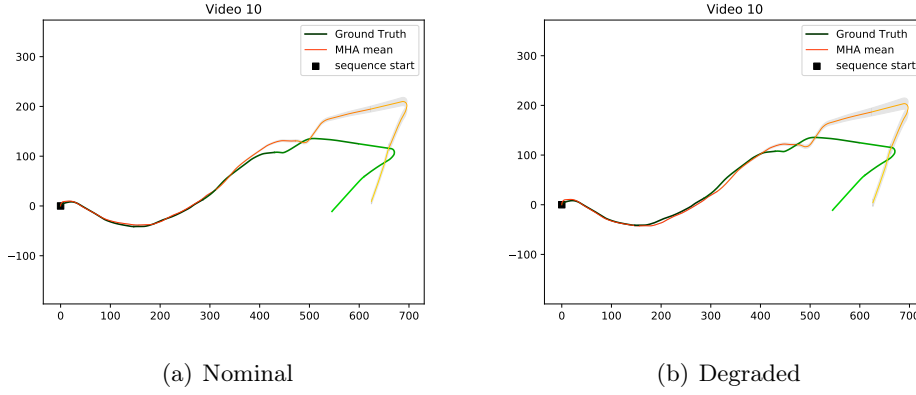


Figure 4.8: Uncertainty in some poses of trajectories for Sequence 10

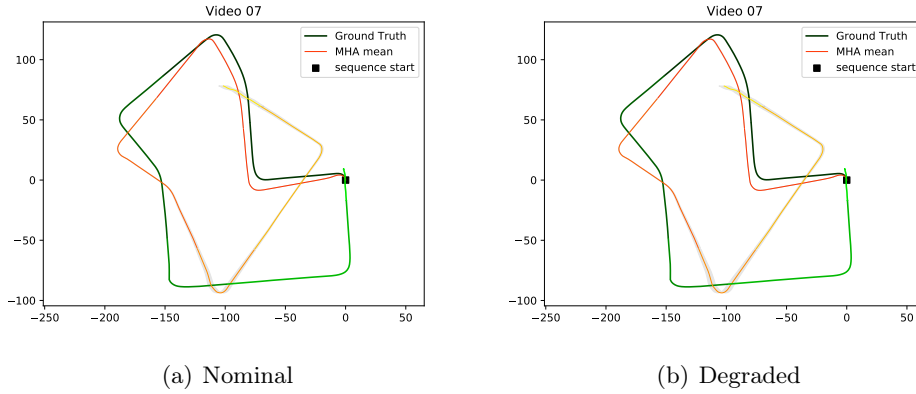


Figure 4.9: Uncertainty in some poses of trajectories for Sequence 07

To further evaluate the behavioural effects of uncertainty, some more representative results can be seen in Fig. 4.10. The errors and uncertainties are shown for nominal and occlusion degradation cases. For Sequence 10 it was observed that the pose error was large along Z-axis as compared to others. A large uncertainty can be seen along this axis which suggests that the error fluctuations can be better captured by it. The box plots show steady increase in uncertainty along with error (strong correlation with error). Higher uncertainty is observed for degraded case in all plots; peak uncertainty occurs for larger rotations (sharp turning motion). This scenario has been depicted in Fig. 4.10g.

4 Experiments, Results and Discussions

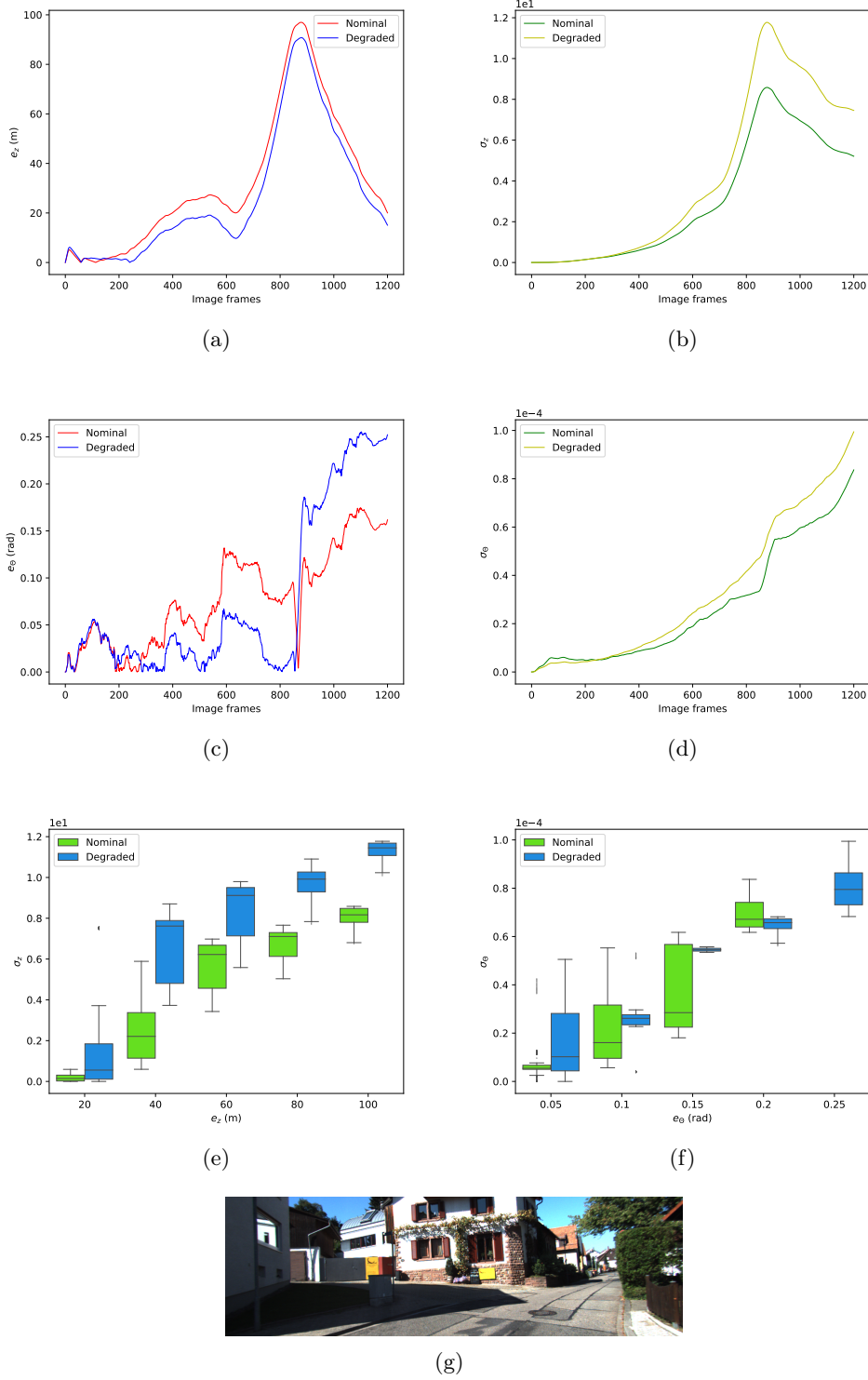


Figure 4.10: (a) and (b) show translational error along z direction and its uncertainty σ_z . (c) and (d) show pitch angle θ error and its uncertainty σ_θ . (e) and (f) show box plots of uncertainty vs. errors for z and θ . (g) shows the scenario where peak error occurs for Sequence 10.

5 Conclusion

In this work, we presented an end-to-end trainable sensor fusion framework for VIO. The framework is independent of the conventional modules of the VIO algorithms without the need for sensor parameters (intrinsic as well as extrinsic). The sensor fusion was carried out using an elaborate representative strategy: multi-head self-attention mechanism, in which the model learns to attend to important features from the visual and inertial feature concatenation. A comparison study was made with the existing state-of-the-art works and deeper insights in regard to complementarity of the sensor modalities were obtained with extensive experiments. The behaviour of all the approaches was analysed for various cases of plausible sensor degradations, showing that robustness of these approaches steadily improves with the selection of a better and more expressive sensor fusion strategy.

We then made our model capable of making predictions with an indication of their certainty. Specifically, the model generated poses with corresponding variance. For this we integrated the deterministic model to a Bayesian Neural Network and performed inference using Laplace approximation without making changes to the model. The uncertainty estimates obtained from Laplace approximation were found to correlate well with error and depict network’s uncertainty when presented with corrupted sensor data. The experimental results implied that incorporating uncertainty information could improve the resilience to such degradations.

There are many aspects that need to be explored and addressed in future research. The main ideas need to be tested on other public datasets in order to verify the generalization ability of the proposed method. For hyperparameter selection, Bayesian optimization using certain learning algorithms[7] can be utilized to obtain more suitable values. In addition, Laplace approximation could be applied to LSTM to study its effect and contribution in uncertainty estimation. Lastly, the interpretability of the proposed fusion strategy could be studied by having a deeper look into what’s actually happening inside the network and if Bayesian treatment helps in improving sensor fusion.

List of Figures

1.1	Cable-suspended aerial manipulator SAM [55]	2
2.1	Example of a network with many convolutional layers [45], filters are applied to each training image at different resolutions, and the output of each convolved image is used as the input to the next layer	7
2.2	FlowNet Simple [16] © 2015 IEEE	8
2.3	Recurrent Neural Network with loop [13]	8
2.4	Folded and unfolded LSTMs and internal structure of its unit [63]. ⊙ and ⊕ denote element-wise product and addition of two vectors, respectively. © 2017 IEEE	10
2.5	An illustration of early fusion, late fusion, and intermediate fusion methods [20]	11
3.1	An architectural overview of the proposed end-to-end VIO framework with multi-head attention mechanism for sensor fusion. σ^2 denotes the variance, i.e., uncertainty of the poses. Image credit: KITTI dataset. (Adapted from [10])	15
4.1	Soft fusion [10]	26
4.2	Predicted trajectories of the KITTI dataset	28
4.3	Predicted trajectories of the KITTI dataset for occlusion degradation	29
4.4	Predicted trajectories of the KITTI dataset for IMU degradation	30
4.5	Predicted trajectories of the KITTI dataset for all vision degradation	31
4.6	Predicted trajectories of the KITTI dataset for all sensor degradation	31
4.7	Soft fusion mask weights for nominal and occlusion degradation case	33
4.8	Uncertainty in some poses of trajectories for Sequence 10	35
4.9	Uncertainty in some poses of trajectories for Sequence 07	35
4.10	(a) and (b) show translational error along z direction and its uncertainty σ_z . (c) and (d) show pitch angle θ error and its uncertainty σ_θ . (e) and (f) show box plots of uncertainty vs. errors for z and θ . (g) shows the scenario where peak error occurs for Sequence 10.	36

List of Tables

3.1	CNN configuration [63] © 2017 IEEE.	16
4.1	Comparison metric for nominal case for sequences 04, 06, 09, 10 . . .	28
4.2	Comparison metric for occlusion vision degradation case	29
4.3	Comparison metric for IMU degradation case	30
4.4	Comparison metric for all vision degradation case	31
4.5	Comparison metric for all sensor degradation case	32

List of Formulas

2.1	Definition: Recurrent Neural Network	9
2.2	Definition: LSTM	9
2.3	Definition: Self-attention	12
2.4	Definition: Multi-head self-attention	12
2.5	Definition: Bayes' Theorem	13
2.6	Definition: Posterior predictive distribution	13
2.7	Definition: Laplace	14
3.1	Definition: Visual feature extractor	16
3.2	Definition: Inertial feature extractor	17
3.3	Definition: Fusion function	17
3.4	Definition: Pose regression	17
3.5	Equation: Conditional probability	18
3.6	Equation: Maximizing conditional probability	18
3.8	Equation: MSE Loss	18
4.1	Equation: Soft mask	26
4.2	Equation: Soft fusion	26
4.3	Definition: Evaluation metrics	27
4.4	Definition: Multiplicative interactions	33

Bibliography

- [1] ALMALIOGLU, Yasin ; TURAN, Mehmet ; SARI, Alp E. ; SAPUTRA, Muhamad Risqi U. ; GUSMÃO, Pedro P. de ; MARKHAM, Andrew ; TRIGONI, Niki: Self-VIO: Self-Supervised Deep Monocular Visual-Inertial Odometry and Depth Estimation. In: *arXiv preprint arXiv:1911.09968* (2019)
- [2] ATREY, Pradeep K. ; HOSSAIN, M A. ; EL SADDIK, Abdulmotaleb ; KANKANHALLI, Mohan S.: Multimodal fusion for multimedia analysis: a survey. In: *Multimedia systems* 16 (2010), Nr. 6, S. 345–379
- [3] AZEVEDO-FILHO, Adriano ; SHACHTER, Ross D.: Laplace’s Method Approximations for Probabilistic Inference in Belief Networks with Continuous Variables. In: MANTARAS], Ramon L. [de (Hrsg.) ; POOLE, David (Hrsg.): *Uncertainty Proceedings 1994*. San Francisco (CA) : Morgan Kaufmann, 1994, S. 28 – 36. – URL <http://www.sciencedirect.com/science/article/pii/B9781558603325500092>. – ISBN 978-1-55860-332-5
- [4] BAHDANAU, Dzmitry ; CHO, Kyunghyun ; BENGIO, Yoshua: Neural machine translation by jointly learning to align and translate. In: *arXiv preprint arXiv:1409.0473* (2014)
- [5] BALTRUŠAITIS, Tadas ; AHUJA, Chaitanya ; MORENCY, Louis-Philippe: Multimodal machine learning: A survey and taxonomy. In: *IEEE transactions on pattern analysis and machine intelligence* 41 (2018), Nr. 2, S. 423–443
- [6] BARBER, David: *Bayesian reasoning and machine learning*. Cambridge University Press, 2012
- [7] BERGSTRA, James S. ; BARDENET, Rémi ; BENGIO, Yoshua ; KÉGL, Balázs: Algorithms for hyper-parameter optimization. In: *Advances in Neural Information Processing Systems*, 2011, S. 2546–2554
- [8] BISHOP, Christopher M.: *Pattern Recognition and Machine Learning*. Springer, 2006
- [9] CHEN, Changhao ; LU, Xiaoxuan ; MARKHAM, Andrew ; TRIGONI, Niki: Ionet: Learning to cure the curse of drift in inertial odometry. In: *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018

- [10] CHEN, Changhao ; ROSA, Stefano ; MIAO, Yishu ; LU, Chris X. ; WU, Wei ; MARKHAM, Andrew ; TRIGONI, Niki: Selective Sensor Fusion for Neural Visual-Inertial Odometry. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019
- [11] CLARK, Ronald ; WANG, Sen ; MARKHAM, Andrew ; TRIGONI, Niki ; WEN, Hongkai: VidLoc: A Deep Spatio-Temporal Model for 6-DoF Video-Clip Relocalization. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017
- [12] CLARK, Ronald ; WANG, Sen ; WEN, Hongkai ; MARKHAM, Andrew ; TRIGONI, Niki: VINet: Visual-Inertial Odometry as a Sequence-to-Sequence Learning Problem. In: *The Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*, 2017
- [13] COLAH: Understanding LSTM Networks. (2015). – URL <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>. – [Online; accessed May 06, 2020]
- [14] COUZINIÉ-DEVY, F. ; SUN, J. ; ALAHARI, K. ; PONCE, J.: Learning to Estimate and Remove Non-uniform Image Blur. In: *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 2013, S. 1075–1082
- [15] D’MELLO, Sidney K. ; KORY, Jacqueline: A review and meta-analysis of multimodal affect detection systems. In: *ACM Computing Surveys (CSUR)* 47 (2015), Nr. 3, S. 1–36
- [16] DOSOVITSKIY, Alexey ; FISCHER, Philipp ; ILG, Eddy ; HAUSSEER, Philip ; HAZIRBAS, Caner ; GOLKOV, Vladimir ; SMAGT, Patrick van der ; CREMERS, Daniel ; BROX, Thomas: FlowNet: Learning Optical Flow With Convolutional Networks. In: *The IEEE International Conference on Computer Vision (ICCV)*, December 2015
- [17] EL-SHEIMY, Naser ; HOU, Haiying ; NIU, Xiaoji: Analysis and modeling of inertial sensors using Allan variance. In: *IEEE Transactions on instrumentation and measurement* 57 (2007), Nr. 1, S. 140–149
- [18] ENGEL, J. ; KOLTUN, V. ; CREMERS, D.: Direct Sparse Odometry. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40 (2018), March, Nr. 3, S. 611–625. – ISSN 1939-3539
- [19] ENGEL, Jakob ; SCHÖPS, Thomas ; CREMERS, Daniel: LSD-SLAM: Large-Scale Direct Monocular SLAM. In: FLEET, David (Hrsg.) ; PAJDLA, Tomas (Hrsg.) ; SCHIELE, Bernt (Hrsg.) ; TUYTELAARS, Tinne (Hrsg.): *Computer Vision – ECCV 2014*. Cham : Springer International Publishing, 2014, S. 834–849. – ISBN 978-3-319-10605-2

- [20] FENG, D. ; HAASE-SCHÄTZ, C. ; ROSENBAUM, L. ; HERTLEIN, H. ; GLÄSNER, C. ; TIMM, F. ; WIESBECK, W. ; DIETMAYER, K.: Deep Multi-Modal Object Detection and Semantic Segmentation for Autonomous Driving: Datasets, Methods, and Challenges. In: *IEEE Transactions on Intelligent Transportation Systems* (2020), S. 1–20
- [21] FORSTER, Christian ; CARLONE, Luca ; DELLAERT, Frank ; SCARAMUZZA, Davide: On-Manifold Preintegration for Real-Time Visual-Inertial Odometry. In: *IEEE Transactions on Robotics* 33 (2016), Nr. 1, S. 1–21
- [22] GEIGER, Andreas ; LENZ, Philip ; STILLER, Christoph ; URTASUN, Raquel: Vision meets Robotics: The KITTI Dataset. In: *International Journal of Robotics Research (IJRR)* (2013)
- [23] GEIGER, Andreas ; LENZ, Philip ; URTASUN, Raquel: Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012
- [24] GITUMA, Mark: What is Optical Flow and why does it matter in deep learning. (2019). – URL <https://medium.com/swlh/what-is-optical-flow-and-why-does-it-matter-in-deep-learning-b3278bb205b5>. – [Online; accessed May 25, 2020]
- [25] GOODFELLOW, Ian J. ; SHLENS, Jonathon ; SZEGEDY, Christian: Explaining and harnessing adversarial examples. In: *arXiv preprint arXiv:1412.6572* (2014)
- [26] HAN, Liming ; LIN, Yimin ; DU, Guoguang ; LIAN, Shiguo: *DeepVIO: Self-supervised Deep Learning of Monocular Visual Inertial Odometry using 3D Geometric Constraints*. 2019
- [27] HEO, Jay ; LEE, Hae B. ; KIM, Saehoon ; LEE, Juho ; KIM, Kwang J. ; YANG, Eunho ; HWANG, Sung J.: Uncertainty-Aware Attention for Reliable Interpretation and Prediction. In: *Advances in Neural Information Processing Systems 31*, Curran Associates, Inc., 2018, S. 909–918. – URL <http://papers.nips.cc/paper/7370-uncertainty-aware-attention-for-reliable-interpretation-and-prediction.pdf>
- [28] HOCHREITER, Sepp ; SCHMIDHUBER, Jürgen: Long short-term memory. In: *Neural computation* 9 (1997), Nr. 8, S. 1735–1780
- [29] HUMT, Matthias: *Laplace Approximation for Uncertainty Estimation of Deep Neural Networks*, Technical University of Munich, Diplomarbeit, 2019. – URL <https://elib.dlr.de/131938/1/thesis.pdf>
- [30] JAYAKUMAR, Siddhant M. ; CZARNECKI, Wojciech M. ; MENICK, Jacob ; SCHWARZ, Jonathan ; RAE, Jack ; OSINDERO, Simon ; TEH, Yee W. ; HARLEY, Tim ; PASCANU, Razvan: Multiplicative Interactions and Where

- to Find Them. In: *International Conference on Learning Representations*, URL <https://openreview.net/forum?id=rylnK6VtDH>, 2020
- [31] KARPATY, A.: Cs231n Convolutional Neural Networks for Visual Recognition. (2020). – URL <https://cs231n.github.io/convolutional-networks/>. – [Online; accessed May 06, 2020]
 - [32] KENDALL, A. ; CIPOLLA, R.: Modelling uncertainty in deep learning for camera relocalization. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, S. 4762–4769. – ISSN null
 - [33] KENDALL, Alex ; CIPOLLA, Roberto: Geometric Loss Functions for Camera Pose Regression With Deep Learning. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017
 - [34] KENDALL, Alex ; GAL, Yarin: What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? In: GUYON, I. (Hrsg.) ; LUXBURG, U. V. (Hrsg.) ; BENGIO, S. (Hrsg.) ; WALLACH, H. (Hrsg.) ; FERGUS, R. (Hrsg.) ; VISHWANATHAN, S. (Hrsg.) ; GARNETT, R. (Hrsg.): *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., 2017, S. 5574–5584. – URL <http://papers.nips.cc/paper/7141-what-uncertainties-do-we-need-in-bayesian-deep-learning-for-computer-vision.pdf>
 - [35] KENDALL, Alex ; GRIMES, Matthew ; CIPOLLA, Roberto: PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization. In: *The IEEE International Conference on Computer Vision (ICCV)*, December 2015
 - [36] KIM, Jaekyum ; CHOI, Jaehyung ; KIM, Yechol ; KOH, Junho ; CHUNG, Chung C. ; CHOI, Jun W.: Robust camera lidar sensor fusion via deep gated information fusion network. In: *2018 IEEE Intelligent Vehicles Symposium (IV)* IEEE (Veranst.), 2018, S. 1620–1625
 - [37] KLEIN, G. ; MURRAY, D.: Parallel Tracking and Mapping for Small AR Workspaces. In: *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, Nov 2007, S. 225–234. – ISSN null
 - [38] LAPLACE, Pierre S.: Mémoire sur la probabilité des causes par les événements (1774). In: *Œuvres compl* (1891), S. 27–65
 - [39] LEE, Jongseok ; BALACHANDRAN, Ribin ; SARKISOV, Yuri S. ; DE STEFANO, Marco ; COELHO, Andre ; SHINDE, Kashmira ; KIM, Min J. ; TRIEBEL, Rudolph ; KONDAK, Konstantin: Visual-inertial telepresence for aerial manipulation. In: *arXiv preprint arXiv:2003.11509* (2020)
 - [40] LEUTENEGGER, Stefan ; LYNEN, Simon ; BOSSE, Michael ; SIEGWART, Roland ; FURGALE, Paul: Keyframe-based visual-inertial odometry using nonlinear optimization. In: *The International Journal of Robotics Research* 34 (2015), Nr. 3, S. 314–334. – URL <https://doi.org/10.1177/0278364914554813>

- [41] LI, Mingyang ; MOURIKIS, Anastasios I.: High-precision, consistent EKF-based visual-inertial odometry. In: *The International Journal of Robotics Research* 32 (2013), Nr. 6, S. 690–711. – URL <https://doi.org/10.1177/0278364913481251>
- [42] LI, R. ; WANG, S. ; LONG, Z. ; GU, D.: UnDeepVO: Monocular Visual Odometry Through Unsupervised Deep Learning. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, S. 7286–7291. – ISSN 2577-087X
- [43] LUONG, Minh-Thang ; PHAM, Hieu ; MANNING, Christopher D.: Effective approaches to attention-based neural machine translation. In: *arXiv preprint arXiv:1508.04025* (2015)
- [44] MACKAY, David J.: A practical Bayesian framework for backpropagation networks. In: *Neural computation* 4 (1992), Nr. 3, S. 448–472
- [45] MATHWORKS, INC.: *Example of a network with many convolutional layers. Filters are applied to each training image at different resolutions, and the output of each convolved image is used as the input to the next layer.* – URL https://www.mathworks.com/solutions/deep-learning/convolutional-neural-network/_jcr_content/mainParsys/band_copy_copy_14735_1026954091/mainParsys/columns_1606542234_c/2/image.adapt.full.high.jpg/1575485682772.jpg. – [Digital Image; accessed April 06, 2020]
- [46] MOURIKIS, A. I. ; ROUMELIOTIS, S. I.: A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation. In: *Proceedings 2007 IEEE International Conference on Robotics and Automation*, April 2007, S. 3565–3572. – ISSN 1050-4729
- [47] MUR-ARTAL, R. ; MONTIEL, J. M. M. ; TARDÓS, J. D.: ORB-SLAM: A Versatile and Accurate Monocular SLAM System. In: *IEEE Transactions on Robotics* 31 (2015), Oct, Nr. 5, S. 1147–1163. – ISSN 1941-0468
- [48] MURPHY, Kevin P.: *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012. – ISBN 0262018020
- [49] NEAL, Radford M.: *Bayesian learning for neural networks*. Bd. 118. Springer Science & Business Media, 2012
- [50] NEWCOMBE, R. A. ; LOVEGROVE, S. J. ; DAVISON, A. J.: DTAM: Dense tracking and mapping in real-time. In: *2011 International Conference on Computer Vision*, Nov 2011, S. 2320–2327. – ISSN 1550-5499
- [51] PASZKE, Adam ; GROSS, Sam ; MASSA, Francisco ; LERER, Adam ; BRADBURY, James ; CHANAN, Gregory ; KILLEEN, Trevor ; LIN, Zeming ; GIMELSHEIN, Natalia ; ANTIGA, Luca ; DESMAISON, Alban ; KOPF, Andreas ; YANG, Edward ;

- DEVITO, Zachary ; RAISON, Martin ; TEJANI, Alykhan ; CHILAMKURTHY, Sasank ; STEINER, Benoit ; FANG, Lu ; BAI, Junjie ; CHINTALA, Soumith: PyTorch: An Imperative Style, High-Performance Deep Learning Library. In: WALLACH, H. (Hrsg.) ; LAROCHELLE, H. (Hrsg.) ; BEYGELZIMER, A. (Hrsg.) ; ALCHÉ-BUC, F. d'(Hrsg.) ; FOX, E. (Hrsg.) ; GARNETT, R. (Hrsg.): *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, S. 8024–8035. – URL <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>
- [52] QIN, T. ; LI, P. ; SHEN, S.: VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator. In: *IEEE Transactions on Robotics* 34 (2018), Aug, Nr. 4, S. 1004–1020. – ISSN 1941-0468
- [53] RITTER, Hippolyt ; BOTEV, Aleksandar ; BARBER, David: A scalable laplace approximation for neural networks. In: *6th International Conference on Learning Representations, ICLR 2018-Conference Track Proceedings* Bd. 6, 2018
- [54] RUGGIERO, Fabio ; LIPPIELLO, Vincenzo ; OLLERO, Anibal: Aerial manipulation: A literature review. In: *IEEE Robotics and Automation Letters* 3 (2018), Nr. 3, S. 1957–1964
- [55] SARKISOV, Yuri S. ; KIM, Min J. ; BICEGO, Davide ; TSETSERUKOU, Dmitry ; OTT, Christian ; FRANCHI, Antonio ; KONDAK, Konstantin: Development of SAM: Cable-Suspended aerial manipulator. In: *2019 International Conference on Robotics and Automation (ICRA)* IEEE (Veranst.), 2019, S. 5323–5329
- [56] SAVAGE, Paul G.: Strapdown Inertial Navigation Integration Algorithm Design Part 1: Attitude Algorithms. In: *Journal of Guidance, Control, and Dynamics* 21 (1998), Nr. 1, S. 19–28. – URL <https://doi.org/10.2514/2.4228>
- [57] SCARAMUZZA, D. ; FRAUNDORFER, F.: Visual Odometry [Tutorial]. In: *IEEE Robotics Automation Magazine* 18 (2011), Dec, Nr. 4, S. 80–92. – ISSN 1558-223X
- [58] SHAMWELL, E. J. ; LEUNG, S. ; NOTHWANG, W. D.: Vision-Aided Absolute Trajectory Estimation Using an Unsupervised Deep Network with Online Error Correction. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2018, S. 2524–2531. – ISSN 2153-0858
- [59] SHU, Yuanchao ; SHIN, Kang G. ; HE, Tian ; CHEN, Jiming: Last-Mile Navigation Using Smartphones. In: *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. New York, NY, USA : Association for Computing Machinery, 2015 (MobiCom '15), S. 512–524. – URL <https://doi.org/10.1145/2789168.2790099>. – ISBN 9781450336192
- [60] SUTSKEVER, Ilya ; VINYALS, Oriol ; LE, Quoc V.: Sequence to Sequence Learning with Neural Networks. In: GHAHRAMANI, Z. (Hrsg.) ; WELLING,

- M. (Hrsg.) ; CORTES, C. (Hrsg.) ; LAWRENCE, N. D. (Hrsg.) ; WEINBERGER, K. Q. (Hrsg.): *Advances in Neural Information Processing Systems 27*. Curran Associates, Inc., 2014, S. 3104–3112. – URL <http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf>
- [61] VASWANI, Ashish ; SHAZEER, Noam ; PARMAR, Niki ; USZKOREIT, Jakob ; JONES, Llion ; GOMEZ, Aidan N. ; KAISER, Lukasz ; POLOSUKHIN, Illia: Attention Is All You Need. In: *CoRR* abs/1706.03762 (2017). – URL <http://arxiv.org/abs/1706.03762>
- [62] WALCH, Florian ; HAZIRBAS, Caner ; LEAL-TAIXE, Laura ; SATTTLER, Torsten ; HILSENBECK, Sebastian ; CREMERS, Daniel: Image-Based Localization Using LSTMs for Structured Feature Correlation. In: *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017
- [63] WANG, Sen ; CLARK, Ronald ; WEN, Hongkai ; TRIGONI, Niki: DeepVO: Towards end-to-end visual odometry with deep Recurrent Convolutional Neural Networks. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, S. 2043–2050
- [64] WANG, Sen ; CLARK, Ronald ; WEN, Hongkai ; TRIGONI, Niki: End-to-end, sequence-to-sequence probabilistic visual odometry through deep neural networks. In: *The International Journal of Robotics Research* 37 (2018), Nr. 4–5, S. 513–542. – URL <https://doi.org/10.1177/0278364917734298>
- [65] WANG, Ting-Chun ; EFROS, Alexei A. ; RAMAMOORTHY, Ravi: Occlusion-aware depth estimation using light-field cameras. In: *Proceedings of the IEEE International Conference on Computer Vision*, 2015, S. 3487–3495
- [66] XU, Kelvin ; BA, Jimmy ; KIROS, Ryan ; CHO, Kyunghyun ; COURVILLE, Aaron ; SALAKHUDINOV, Ruslan ; ZEMEL, Rich ; BENGIO, Yoshua: Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In: BACH, Francis (Hrsg.) ; BLEI, David (Hrsg.): *Proceedings of the 32nd International Conference on Machine Learning* Bd. 37. Lille, France : PMLR, 07–09 Jul 2015, S. 2048–2057. – URL <http://proceedings.mlr.press/v37/xuc15.html>
- [67] YIN, Zhichao ; SHI, Jianping: GeoNet: Unsupervised Learning of Dense Depth, Optical Flow and Camera Pose. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018
- [68] ZAMIR, Amir R. ; WEKEL, Tilman ; AGRAWAL, Pulkit ; WEI, Colin ; MALIK, Jitendra ; SAVARESE, Silvio: Generic 3d representation via pose estimation and matching. In: *European Conference on Computer Vision* Springer (Veranst.), 2016, S. 535–553
- [69] ZHAN, Huangying ; GARG, Ravi ; SAROJ WEERASEKERA, Chamara ; LI, Kejie ; AGARWAL, Harsh ; REID, Ian: Unsupervised Learning of Monocular Depth

Estimation and Visual Odometry With Deep Feature Reconstruction. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018

- [70] ZHOU, Tinghui ; BROWN, Matthew ; SNAVELY, Noah ; LOWE, David G.: Un-supervised Learning of Depth and Ego-Motion From Video. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017