

# Selective Sensor Fusion for Neural Visual-Inertial Odometry

Changhao Chen<sup>1</sup>, Stefano Rosa<sup>1</sup>, Yishu Miao<sup>2</sup>, Chris Xiaoxuan Lu<sup>1</sup>,  
Wei Wu<sup>3</sup>, Andrew Markham<sup>1</sup>, Niki Trigoni<sup>1</sup>

<sup>1</sup>Department of Computer Science, University of Oxford

<sup>2</sup>MO Intelligence <sup>3</sup>Tencent

## Abstract

*Deep learning approaches for Visual-Inertial Odometry (VIO) have proven successful, but they rarely focus on incorporating robust fusion strategies for dealing with imperfect input sensory data. We propose a novel end-to-end selective sensor fusion framework for monocular VIO, which fuses monocular images and inertial measurements in order to estimate the trajectory whilst improving robustness to real-life issues, such as missing and corrupted data or bad sensor synchronization. In particular, we propose two fusion modalities based on different masking strategies: deterministic soft fusion and stochastic hard fusion, and we compare with previously proposed direct fusion baselines. During testing, the network is able to selectively process the features of the available sensor modalities and produce a trajectory at scale. We present a thorough investigation on the performances on three public autonomous driving, Micro Aerial Vehicle (MAV) and hand-held VIO datasets. The results demonstrate the effectiveness of the fusion strategies, which offer better performances compared to direct fusion, particularly in presence of corrupted data. In addition, we study the interpretability of the fusion networks by visualising the masking layers in different scenarios and with varying data corruption, revealing interesting correlations between the fusion networks and imperfect sensory input data.*

## 1. Introduction

Humans are able to perceive their self-motion through space via multimodal perceptions. Optical flow (visual cues) and vestibular signals (inertial motion sense) are the two most sensitive cues for determining self-motion [9].

In the fields of computer vision and robotics, integrating visual and inertial information in the form of Visual-Inertial Odometry (VIO) is a well researched topic [17, 20, 19, 11, 29], as it enables ubiquitous mobility for mobile agents by providing robust and accurate pose information. Moreover, cameras and inertial sensors are relatively low-cost, power-

efficient and widely found in ground robots, smartphones, and unmanned aerial vehicles (UAVs). Existing VIO approaches generally follow a standard pipeline that involves fine-tuning of both feature detection and tracking, and of the sensor fusion strategy. These models rely on handcrafted features, and fuse the information based on filtering [20] or nonlinear optimization [19, 11, 29]. However, naively using all features before fusion will lead to unreliable state estimation, as incorrect feature extraction or matching cripples the entire system. Real issues which can and do occur include camera occlusion or operation in low-light conditions [40], excess noise or drift within the inertial sensor [26], time-synchronization between the two streams or spatial misalignment [21].

Recent studies on applying deep neural networks (DNNs) to solving visual-inertial odometry [30] or visual odometry [18, 7] showed competitive performance in terms of both accuracy and robustness. Although DNNs excel at extracting high-level features representative of egomotion, these learning-based methods are not explicitly modelling the sources of degradation in real-world usages. Without considering possible sensor errors, all features are directly fed into other modules for further pose regression in [4, 7, 18], or simply concatenated as in [30]. These factors can possibly cause troubles to the accuracy and safety of VIO systems, when the input data are corrupted or missing.

For this reason, we present a generic framework that models feature selection for robust sensor fusion. The selection process is conditioned on the measurement reliability and the dynamics of both egomotion and environment. Two alternative feature weighting strategies are presented: soft fusion, implemented in a deterministic fashion; and hard fusion, which introduces stochastic noise and intuitively learns to keep the most relevant feature representations, while discarding useless or misleading information. Both architectures are trained in an end-to-end fashion.

By explicitly modelling the selection process, we are able to demonstrate the strong correlation between the selected features and the environmental/measurement dynamics by visualizing the sensor fusion masks, as illus-

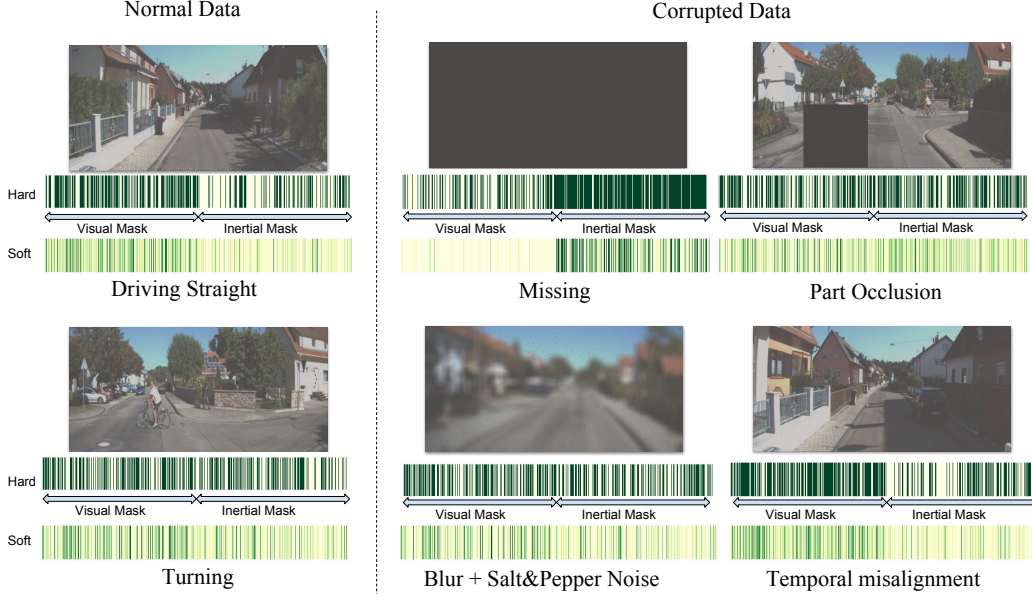


Figure 1: Visualization of the learned hard and soft fusion masks under different conditions (left: normal data; middle and right: corrupted data). The number (hard) or weights (soft) of selected features in the visual and inertial sides can reflect the self-motion dynamics (increasing importance of inertial features during turning), and data corruption conditions.

trated in Figure 1. Our results show that features extracted from different modalities (i.e., vision and inertial motion) are complementary in various conditions: the inertial features contribute more in presence of fast rotation, while visual features are preferred during large translations (Figure 6). Thus, the selective sensor fusion provides insight into the underlying strengths of each sensor modality. We also demonstrate how incorporating selective sensor fusion makes VIO robust to data corruption typically encountered in real-world scenarios.

The main contributions of this work are as follows:

- We present a generic framework to learn selective sensor fusion enabling more robust and accurate ego-motion estimation in real world scenarios.
- Our selective sensor fusion masks can be visualized and interpreted, providing deeper insight into the relative strengths of each stream, and guiding further system design.
- We create challenging datasets on top of current public VIO datasets by considering seven different sources of sensor degradation, and conduct a new and complete study on the accuracy and robustness of deep sensor fusion in presence of corrupted data.

## 2. Neural VIO Models with Selective Fusion

In this section, we introduce the end-to-end architecture for neural visual-inertial odometry, which is the foundation

for our proposed framework. Figure 2(top) shows a modular overview of the architecture, consisting of visual and inertial encoders, feature fusion, temporal modelling and pose regression. Our model takes in a sequence of raw images and IMU measurements, and generates their corresponding pose transformation. With the exception of our novel feature fusion, the pipeline can be any generic deep VIO technique. In the Feature Fusion component we propose two different selection mechanisms (soft and hard) and compare them with direct (i.e. a uniform/unweighted mask) fusion, as shown in Figure 2(bottom).

### 2.1. Feature Encoder

**Visual Feature Encoder** The Visual Encoder extracts a latent representation from a set of two consecutive monocular images  $\mathbf{x}_V$ . Ideally, we want the Visual Encoder  $f_{\text{vision}}$  to learn geometrically meaningful features rather than features related with appearance or context. For this reason, instead of using a PoseNet model [18], as commonly found in other DL-based VO approaches [43, 42, 41], we use FlowNetSimple [10] as our feature encoder. FlowNet provides features that are suited for optical flow prediction. The network consists of nine convolutional layers. The size of the receptive fields gradually reduces from  $7 \times 7$  to  $5 \times 5$  and finally  $3 \times 3$ , with stride two for the first six. Each layer is followed by a ReLU nonlinearity except for the last one, and we use the features from the last convolutional layer  $\mathbf{a}_V$  as our visual feature:

$$\mathbf{a}_V = f_{\text{vision}}(\mathbf{x}_V). \quad (1)$$

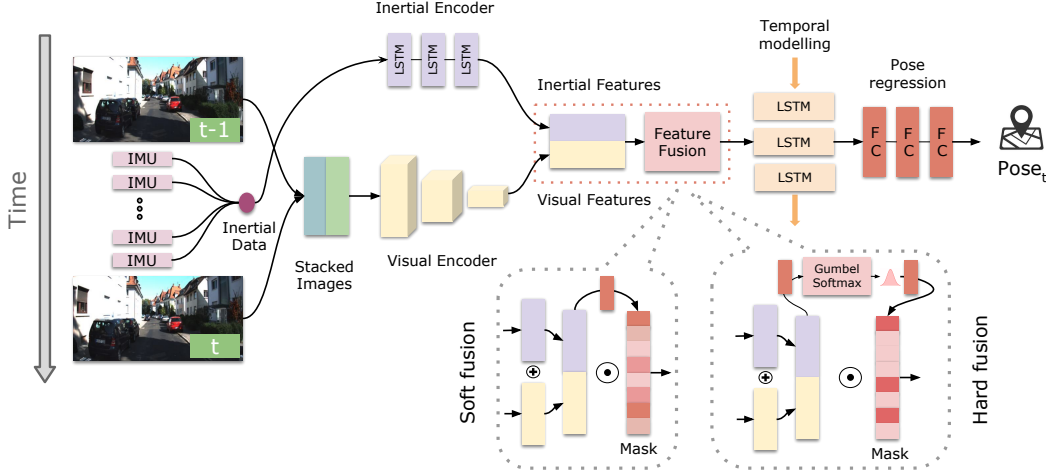


Figure 2: An overview of our neural visual-inertial odometry architecture with proposed selective sensor fusion, consisting of visual and inertial encoders, feature fusion, temporal modelling and pose regression. In the feature fusion component, we compare our proposed soft and hard selective sensor fusion strategies with direct fusion.

**Inertial Feature Encoder:** Inertial data streams have a strong temporal component, and are generally available at higher frequency ( $\sim 100$  Hz) than images ( $\sim 10$  Hz). Inspired by IONet [6], we use a two-layer Bi-directional LSTM with 128 hidden states as the Inertial Feature Encoder  $f_{\text{inertial}}$ . As shown in Figure 2, a window of inertial measurements  $\mathbf{x}_I$  between each two images is fed to the inertial feature encoder in order to extract the dimensional feature vector  $\mathbf{a}_I$ :

$$\mathbf{a}_I = f_{\text{inertial}}(\mathbf{x}_I). \quad (2)$$

## 2.2. Fusion Function

We now combine the high-level features produced by the two encoders from raw data sequences, with a fusion function  $g$  that combines information from the visual  $\mathbf{a}_V$  and inertial  $\mathbf{a}_I$  channels to extract the useful combined feature  $\mathbf{z}$  for future pose regression task:

$$\mathbf{z} = g(\mathbf{a}_V, \mathbf{a}_I). \quad (3)$$

There are several different ways to implement this fusion function. The current approach is to directly concatenate the two features together into one feature space (we call this method direct fusion  $g_{\text{direct}}$ ). However, in order to learn a robust sensor fusion model, we propose two fusion schemes – deterministic soft fusion  $g_{\text{soft}}$  and stochastic hard fusion  $g_{\text{hard}}$ , which explicitly model the feature selection process according to the current environment dynamics and the reliability of the data input. Our selective fusion mechanisms re-weights the concatenated inertial-visual features, guided by the concatenated features themselves. The fusion network is another deep neural network and is end-to-end trainable. Details will be discussed in Section 3.

## 2.3. Temporal Modelling and Pose Regression

The fundamental tenet of ego-motion estimation requires modeling temporal dependencies to derive accurate pose regression. Hence, a recurrent neural network (a two-layer Bi-directional LSTM) takes in input the combined feature representation  $\mathbf{z}_t$  at time step  $t$  and its previous hidden states  $\mathbf{h}_{t-1}$  and models the dynamics and connections between a sequence of features. After the recurrent network, a fully-connected layer serves as the pose regressor, mapping the features to a pose transformation  $\mathbf{y}_t$ , representing the motion transformation over a time window.

$$\mathbf{y}_t = \text{RNN}(\mathbf{z}_t, \mathbf{h}_{t-1}) \quad (4)$$

## 3. Selective Sensor Fusion

Intuitively, the features from each modality offer different strengths for the task of regressing pose transformations. This is particularly true in the case of visual-inertial odometry (VIO), where the monocular visual input is capable of estimating the appearance and geometry of a 3D scene, but is unable to determine the metric scale [11]. Moreover, changes in illumination, textureless areas and motion blur can lead to bad data association. Meanwhile, inertial data is interoceptive/egocentric and generally environment-agnostic, and can still be reliable when visual tracking fails [6]. However, measurements from low-cost MEMS inertial sensors are corrupted by inevitable noise and bias, which leads to higher long-term drift than a well-functioning visual-odometry chain.

Our perspective is that simply considering all features as though they are correct, without any consideration of degradation, is unwise and will lead to unrecoverable er-

errors. In this section, we propose two different selective sensor fusion schemes for explicitly learning the feature selection process: soft (deterministic) fusion, and hard (stochastic) fusion, as illustrated in Figure 3. In addition, we also present a straightforward sensor fusion scheme – direct fusion – as a baseline model for comparison.

### 3.1. Direct Fusion

A straightforward approach for implementing sensor fusion in a VIO framework consists in the use of Multi-Layer Perceptrons (MLPs) to combine the features from the visual and inertial channels. Ideally, the system learns to perform feature selection and prediction in an end-to-end fashion. Hence, direct fusion is modelled as:

$$g_{\text{direct}}(\mathbf{a}_V, \mathbf{a}_I) = [\mathbf{a}_V; \mathbf{a}_I] \quad (5)$$

where  $[\mathbf{a}_V; \mathbf{a}_I]$  denotes an MLP function that concatenates  $\mathbf{a}_V$  and  $\mathbf{a}_I$ .

### 3.2. Soft Fusion (Deterministic)

We now propose a soft fusion scheme that explicitly and deterministically models feature selection. Similar to the widely applied attention mechanism [33, 39, 14], this function re-weights each feature by conditioning on both the visual and inertial channels, which allows the feature selection process to be jointly trained with other modules. The function is deterministic and differentiable.

Here, a pair of continuous masks  $\mathbf{s}_V$  and  $\mathbf{s}_I$  is introduced to implement soft selection of the extracted feature representations, before these features are passed to temporal modelling and pose regression:

$$\mathbf{s}_V = \text{Sigmoid}_V([\mathbf{a}_V; \mathbf{a}_I]) \quad (6)$$

$$\mathbf{s}_I = \text{Sigmoid}_I([\mathbf{a}_V; \mathbf{a}_I]) \quad (7)$$

where  $\mathbf{s}_V$  and  $\mathbf{s}_I$  are the masks applied to visual features and inertial features respectively, and which are deterministically parameterised by the neural networks, conditioned on both the visual  $\mathbf{a}_V$  and inertial features  $\mathbf{a}_I$ . The sigmoid function makes sure that each of the features will be re-weighted in the range  $[0, 1]$ .

Then, the visual and inertial features are element-wise multiplied with their corresponding soft masks as the new re-weighted vectors. The selective soft fusion function is modelled as

$$g_{\text{soft}}(\mathbf{a}_V, \mathbf{a}_I) = [\mathbf{a}_V \odot \mathbf{s}_V; \mathbf{a}_I \odot \mathbf{s}_I]. \quad (8)$$

### 3.3. Hard Fusion (Stochastic)

In addition to the soft fusion introduced above, we propose a variant of the fusion scheme – hard fusion. Instead of re-weighting each feature by a continuous value, hard fusion learns a stochastic function that generates a binary

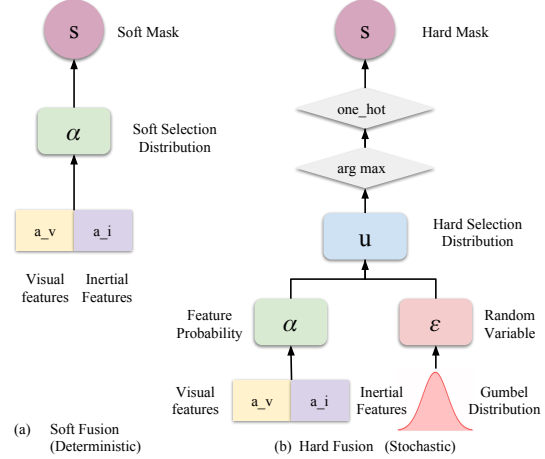


Figure 3: An illustration of our proposed soft (deterministic) and hard (stochastic) feature selection process.

mask that either propagates the feature or blocks it. This mechanism can be viewed as a switcher for each component of the feature map, which is stochastic neural implemented by a parameterised Bernoulli distributions.

However, the stochastic layer cannot be trained directly by back-propagation, as gradients will not propagate through discrete latent variables. To tackle this, the REINFORCE algorithm [38, 24] is generally used to construct the gradient estimator. In our case, we employ a more lightweight method – Gumbel-Softmax resampling [16, 22] to infer the stochastic layer, so that the hard fusion can be trained in an end-to-end fashion as well.

Instead of learning masks deterministically from features, hard masks  $\mathbf{s}_V$  and  $\mathbf{s}_I$  are re-sampled from a Bernoulli distribution, parameterised by  $\alpha$ , which is conditioned on features but with the addition of stochastic noise:

$$\mathbf{s}_V \sim p(\mathbf{s}_V | \mathbf{a}_V, \mathbf{a}_I) = \text{Bernoulli}(\alpha_V) \quad (9)$$

$$\mathbf{s}_I \sim p(\mathbf{s}_I | \mathbf{a}_V, \mathbf{a}_I) = \text{Bernoulli}(\alpha_I). \quad (10)$$

Similar to soft fusion, features are element-wise multiplied with their corresponding hard masks as the new reweighted vectors. The stochastic hard fusion function is modelled as

$$g_{\text{hard}}(\mathbf{a}_V, \mathbf{a}_I) = [\mathbf{a}_V \odot \mathbf{s}_V; \mathbf{a}_I \odot \mathbf{s}_I]. \quad (11)$$

Figure 3 (b) shows the detailed workflow of proposed Gumbel-Softmax resampling based hard fusion. A pair of probability variables  $\alpha_V$  and  $\alpha_I$  is conditioned on the concatenated visual and inertial feature vectors  $[\mathbf{a}_V; \mathbf{a}_I]$ :

$$\alpha_V = \text{Sigmoid}_V([\mathbf{a}_V; \mathbf{a}_I]) \quad (12)$$

$$\alpha_I = \text{Sigmoid}_I([\mathbf{a}_V; \mathbf{a}_I]), \quad (13)$$

where the probability variables are n-dimensional vectors  $\alpha = [\pi_1, \dots, \pi_n]$ , representing the probability of each fea-

ture at location  $n$  to be selected or not. Sigmoid function enables each vector to be re-weighted in the range  $[0, 1]$ .

The Gumbel-max trick [23] allows efficiently to draw samples  $s$  from a categorical distribution given the class probabilities  $\pi_i$  and a random variable  $\epsilon_i$ , and then the one-hot encoding performs "binarization" of the category:

$$s = \text{one\_hot}(\arg \max_i [\epsilon_i + \log \pi_i]). \quad (14)$$

This is due to the fact that for any  $B \subseteq [1, \dots, n]$  [13]:

$$\arg \max_i [\epsilon_i + \log \pi_i] \sim \frac{\pi_i}{\sum_{i \in B} \pi_i} \quad (15)$$

It could be viewed as a process of adding independent Gumbel perturbations  $\epsilon_i$  to the discrete probability variable. In practice, the random variable  $\epsilon_i$  is sampled from a Gumbel distribution, which is a continuous distribution on the simplex that can approximate categorical samples:

$$\epsilon = -\log(-\log(u)), u \sim \text{Uniform}(0, 1). \quad (16)$$

In Equation 14 the argmax operation is not differentiable, so Softmax function is instead used as an approximate:

$$h_i = \frac{\exp((\log(\pi_i) + \epsilon_i)/\tau)}{\sum_{i=1}^n \exp((\log(\pi_j) + \epsilon_j)/\tau)}, i = 1, \dots, n, \quad (17)$$

where  $\tau > 0$  is the temperature that modulates the resampling process.

### 3.4. Discussions on Neural and classical VIOs

Basically, soft fusion gently re-weights each feature in a deterministic way, while hard fusion directly blocks features according to the environment and its reliability. In general, soft fusion is a simple extension of direct fusion that is good for dealing with the uncertainties in the input sensory data. By comparison, the inference in hard fusion is more difficult, but it offers a more intuitive representation. The stochasticity gives the VIO system better generalisation ability and higher tolerance to imperfect sensory data. The stochastic mask of hard fusion acts as an inductive bias, separating the feature selection process from prediction, which can also be easily interpreted by corresponding to uncertainties of the input sensory data.

Filtering methods update their belief based on the past state and current observations of visual and inertial modalities [25, 20, 15, 2]. "Learning" within these methods is usually constrained to gain and covariances [1]. This is a deterministic process, and noise parameters are hand-tuned beforehand. Deep learning methods are instead fully learned from data and the hidden recurrent state only contains information relevant to the regressor. Our approach models the feature selection process explicitly with the use of soft and hard masks. Loosely, the proposed soft mask can be viewed as similar to tuning the gain and covariance matrix in classical filtering methods, but based on the latent data representation instead.

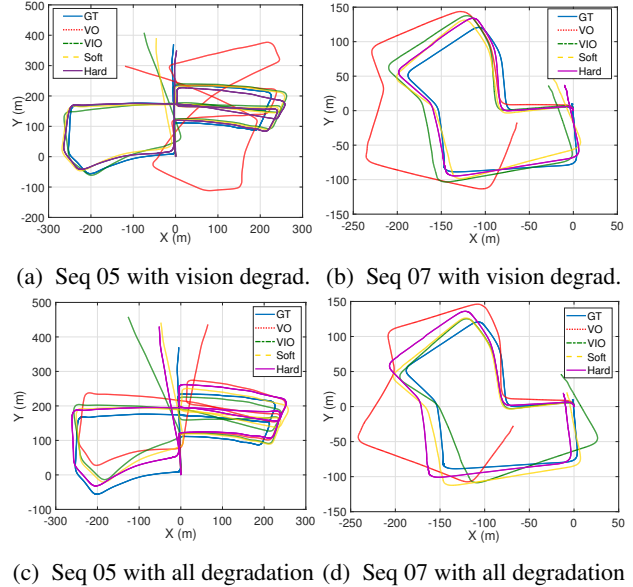


Figure 4: Estimated trajectories on the KITTI dataset. Top row: dataset with vision degradation (10% occlusion, 10% blur, and 10% missing data); bottom row: data with all degradation (5% for each). Here, GT, VO, VIO, Soft and Hard mean the ground truth, neural vision-only model, neural visual inertial models with direct, soft, and hard fusion.

## 4. Experiments

We evaluate our proposed approaches on three well-known datasets: the KITTI Odometry dataset for autonomous driving [12], the EuRoC dataset for micro aerial vehicle [5], and the PennCOSYVIO dataset for hand-held devices [28]. A demonstration video and other details can be found at our [project website](https://changhaoc.github.io/selective_sensor_fusion/)<sup>1</sup>.

### 4.1. Experimental Setup and Baselines

The architecture was implemented with PyTorch and trained on a NVIDIA Titan X GPU.

We chose the neural vision-only model and the neural visual-inertial model with direct fusion as our baselines, termed Vision-Only (DeepVO) and VIO-Direct (VINet) respectively in our experiments. The neural vision-only model uses the visual encoder, temporal modelling and pose regression as in our proposed framework in Figure 2. Neural visual inertial model with direct fusion uses the same framework as in our proposed selective fusion except the feature fusion component. All of the networks including baselines were trained with a batch size of 8 using the Adam optimizer, with a learning rate  $lr = 1e^{-4}$ . The hyper-parameters inside the networks were identical for a fair comparison.

<sup>1</sup>[https://changhaoc.github.io/selective\\_sensor\\_fusion/](https://changhaoc.github.io/selective_sensor_fusion/)

Table 1: Effectiveness of different sensor fusion strategies in presence of different kinds of sensor data corruption. For each case we report absolute translational error (m) and rotational error (degrees).

Model	Vision Degradation			IMU Degradation		Sensor Degradation	
	Occlusion	Blur	Missing	Noise and bias	Missing	Spatial	Temporal
Vision Only	0.117,0.148	0.117,0.153	0.213,0.456	0.116,0.136	0.116,0.136	0.116,0.136	0.116,0.136
VIO Direct	0.116,0.110	0.117,0.107	0.191,0.155	0.118,0.115	0.118,0.163	0.119,0.137	0.120,0.111
VIO Soft	0.116, <b>0.105</b>	0.119, <b>0.104</b>	0.198, <b>0.149</b>	0.119, <b>0.105</b>	0.118, <b>0.129</b>	0.119, <b>0.128</b>	0.119, <b>0.108</b>
VIO Hard	<b>0.112</b> ,0.126	<b>0.114</b> ,0.110	<b>0.187</b> ,0.159	<b>0.114</b> ,0.120	<b>0.115</b> ,0.140	<b>0.111</b> ,0.146	<b>0.113</b> ,0.133

## 4.2. Datasets

**KITTI Odometry dataset** [12] We used Sequences *00, 01, 02, 04, 06, 08, 09* for training and tested the network on Sequences *05, 07, and 10*, excluding sequence 03 as the corresponding raw file is unavailable. The images and ground-truth provided by GPS are collected at 10 Hz, while the IMU data is at 100 Hz.

**EuRoC Micro Aerial Vehicle dataset** [5] It contains tightly synchronized video streams from a Micro Aerial Vehicle (MAV), carrying a stereo camera and an IMU, and is composed by 11 flight trajectories in two environments, exhibiting complex motion. We used Sequence *MH\_04\_difficult* for testing, and left the other sequences for training. We downsampled the images and IMUs to 10 Hz and 100 Hz respectively.

**PennCOSYVIO dataset** [28] It is composed by four sequences where the user is carrying multiple visual and inertial sensors rigidly attached. We used Sequences *bs, as* and *bf* for training, and *af* for testing. The images and IMUs were downsampled to 10 Hz and 100 Hz respectively.

## 4.3. Data Degradation

In order to provide an extensive study of the effects of sensor data degradation and to evaluate the performances of the proposed approach, we generate three categories of degraded datasets, by adding various types of noise and occlusion to the original data, as described in the following subsections.

### 4.3.1 Vision Degradation

**Occlusions:** we overlay a mask of dimensions  $128 \times 128$  pixels on top of the sample images, at random locations for each sample. Occlusions can happen due to dust or dirt on the sensor or stationary objects close to the sensor [37].

**Blur+noise:** we apply Gaussian blur with  $\sigma=15$  pixels to the input images, with additional salt-and-pepper noise. Motion blur and noise can happen when the camera or the light condition changes substantially [8].

**Missing data:** we randomly remove 10% of the input images. This can occur when packets are dropped from the bus due to excess load or temporary sensor disconnection. It can also occur if we pass through an area of very poor

illumination e.g. a tunnel or underpass.

### 4.3.2 IMU Degradation

**Noise+bias:** on top of the already noisy sensor data we add additive white noise to the accelerometer data and a fixed bias on the gyroscope data. This can occur due to increased sensor temperature and mechanical shocks, causing inevitable thermo-mechanical white noise and random walking noise [26].

**Missing data:** we randomly remove windows of inertial samples between two consecutive random visual frames. This can occur when the IMU measuring is unstable or packets are dropped from the bus.

### 4.3.3 Cross-Sensor Degradation

**Spatial misalignment:** we randomly alter the relative rotation between the camera and the IMU, compared to the initial extrinsic calibration. This can occur due to axis misalignment and the incorrect sensor calibration [20]. We uniformly model up to 10 degrees of misalignment .

**Temporal misalignment:** we apply a time shift between windows of input images and windows of inertial measurements. This can happen due to relative drifts in clocks between independent sensor subsystems [21].

## 4.4. Detailed Investigation on Robustness to Data Corruption

Table 1 shows the relative performance of the proposed data fusion strategies, compared with the baselines. In particular, we compare with a DeepVO [36] (Vision-Only) implementation, and finally with an implementation of VINet [30] (VIO Direct), which uses a naïve fusion strategy by concatenating visual and inertial features. Figure 4 shows a visual comparison of the resulting test trajectories in presence of visual and combined degradations. In the vision degraded set the input images are randomly degraded by adding occlusion, blurring+noise and removing images, with 10% probability for each degradation. In the full degradation set, images and IMU sequences from the dataset are corrupted by all seven degradations with a probability of 5% each. As a metric, we always report the average absolute error on relative translation and rotation estimates

Table 2: Results on autonomous driving scenario [12].

	Normal Data	Vision Degradation	All Degradation
Vision Only	0.116,0.136	0.177,0.355	<b>0.142</b> ,0.281
VIO Direct	0.116,0.106	0.175,0.164	0.148,0.139
VIO Soft	0.118, <b>0.098</b>	0.173, <b>0.150</b>	0.152, <b>0.134</b>
VIO Hard	<b>0.112</b> ,0.110	<b>0.172</b> ,0.151	0.145,0.150

Table 3: Results on UAV scenario [5].

	Normal Data	Vision Degradation	All Degradation
Vision Only	0.00976,0.0867	0.0222,0.268	0.0190,0.213
VIO Direct	<b>0.00765</b> , <b>0.0540</b>	0.0181,0.0696	0.0162,0.0935
VIO Soft	0.00848,0.0564	<b>0.0170</b> , <b>0.0533</b>	<b>0.0152</b> ,0.0860
VIO Hard	0.00795,0.0589	0.0177,0.0565	0.0157, <b>0.0823</b>

Table 4: Results on handheld scenario [28].

	Normal Data	Vision Degradation	All Degradation
Vision Only	0.0379,1.755	0.0446,1.849	0.0414,1.875
VIO Direct	<b>0.0377</b> ,1.350	<b>0.0396</b> ,1.223	0.0407,1.353
VIO Soft	0.0381, <b>1.252</b>	0.0399, <b>1.166</b>	0.0405,1.296
VIO Hard	0.0387,1.296	0.0410,1.206	<b>0.0400</b> , <b>1.232</b>

over the trajectory, in order to avoid the shortcomings of approaches using global reference frames to compute errors.

Some interesting behaviours emerge from Table 1. Firstly, as expected, both the proposed fusion approaches outperform VO and the baseline VIO fusion approaches when subject to degradation. Our intuition is that the visual features are likely to be local and discrete, and as such, erroneous regions can be blanked out, which would benefit the fusion network when it is predominantly relying on vision. Conversely, inertial data is continuous and thus a more gradual reweighting as performed by the soft fusion approach would preserve these features better. As inertial data is more important for rotation, this could explain this observation. More interestingly, the soft fusion always improves the angle component estimation, while the hard fusion always improves the translation component estimation.

Table 5: Comparison with classical methods

	Normal data	Full visual degr.	Occl.+blur	Full sensor degr.
KITTI	0.116,0.044	Fail	2.4755,0.0726	Fail
EuRoC	0.0283,0.0402	0.0540,0.0591	0.0198,0.0400	Fail

#### 4.5. Results on autonomous driving, UAV scenario and hand-held scenario

Table 2 shows the aggregate results on the KITTI dataset in presence of normal data, all combined visual degradation and all combined visual+inertial degradation. In particular, we compare with two deep approaches: DeepVO (Vision-Only) and an implementation of VINet (VIO Direct). We can see the same fusion behavior as in Table 1.

Table 3 reports the error results on EuRoC. Similar to KITTI, the soft fusion strategy consistently improves the angle estimation, while the hard fusion always improves the translation estimation. Interestingly, in the hand-held sce-

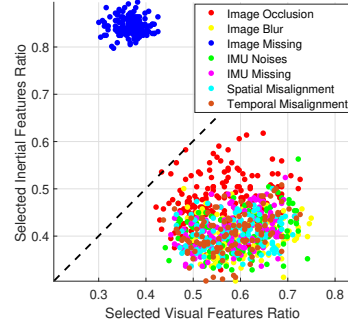


Figure 5: A comparison of visual and inertial features selection rate in seven data degradation scenarios.

nario (Table 4) there is less marked difference between the different fusion strategies regarding the translation component. This can be due to the small size of the dataset and the nature of motion, leading the network to slightly overfit on linear translations. However, hard fusion still improves both errors in presence of both visual and inertial degradation. This could be ascribed to the direct fusion method overfitting on visual data, while a few transitions from outdoor to indoor introduce illumination changes and occlusion.

#### 4.6. Comparison with Classical VIOs

For KITTI, due to the lack of time synchronization between IMUs and images, both OKVIS [19] and VINS-Mono [29] cannot work. We instead provide results from an implementation of MSCKF [15]<sup>2</sup>. For EuRoC MAV we compare with OKVIS [19]<sup>3</sup>.

As shown in Table 5, on KITTI, MSCKF fails with full degradation due to the missing images; on EuRoC OKVIS handles missing images instead but both baselines fail with full sensor degradation due to the temporal misalignment. Learning-based methods reach comparable position/translation errors, but the orientation error is always lower for traditional methods. Because DNNs shine at extracting features and regressing translation from raw images, while IMUs improve filtering methods to get better orientation results on normal data. Interestingly, the performance of learning-based fusion strategies degrade gracefully in the presence of corrupted data, while filtering methods fail abruptly with the presence of large sensor noise and misalignment issues.

#### 4.7. Interpretation of Selective Fusion

Incorporating hard mask into our framework enables us to quantitatively and qualitatively interpret the fusion process. Firstly, we analyse the contribution of each individual modality in different scenarios. Since hard fusion blocks

<sup>2</sup>The code can be found at: <https://uk.mathworks.com/matlabcentral/fileexchange/43218-visual-inertial-odometry>

<sup>3</sup>The code can be found at: <https://github.com/ethz-asl/okvis>

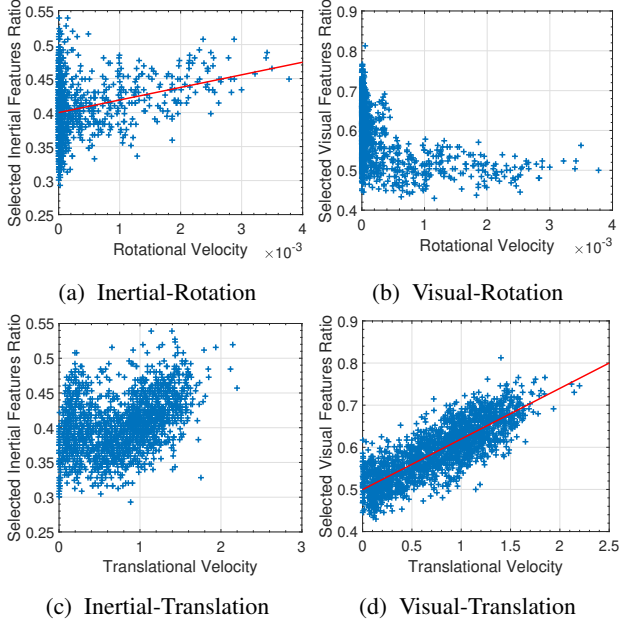


Figure 6: Correlations between the number of inertial/visual features and amount of rotation/translation show that the inertial features contribute more with rotation rates, e.g. turning, while more visual features are selected with increasing linear velocity.

some features according to their reliability, in order to interpret the “feature selection” mechanism we simply compare the ratio of the non-blocked features for each modality. Figure 5 shows that visual features dominate compared with inertial features in most scenarios. Non-blocked visual features are more than 60%, underlining the importance of this modality. We see no obvious change when facing small visual degradation, such as image blur, because the FlowNet extractor can deal with such disturbances. However, when the visual degradation becomes stronger the role of inertial features becomes significant. Notably, the two modalities contribute equally in presence of occlusion. Inertial features dominate with missing images by more than 90%.

In Figure 6 we analyze the correlation between amount of linear and angular velocity and the selected features. These results also show how the belief on inertial features is stronger in presence of large rotations, e.g. turning, while visual features are more reliable with increasing linear translations. It is interesting to see that at low translational velocity (0.5m / 0.1s) only 50% to 60% visual features are activated, while at high speed (1.5m / 0.1s) 60% to 75% visual features are used.

## 5. Related Work

**Visual Inertial Odometry** Traditionally, visual-inertial approaches can be roughly segmented into three different

classes according to their information fusion methods: filtering approaches [17], fixed-lag smoothers [19] and full smoothing methods [11]. In classical VIO models, their features are handcrafted, as OKVIS [19] presented a keyframe-based approach that jointly optimizes visual feature reprojections and inertial error terms. Semi-direct [32] and direct [34] methods have been proposed in an effort to move towards feature-less approaches, removing the feature extraction pipeline for increased speed. Recent VINet [30] used neural network to learn visual-inertial navigation, but only fused two modalities in a naive concatenation way. We provide a generic framework for deep features fusion, and outperformed the direct fusion in different scenarios.

**Deep Neural Networks for Localization** Recent data-driven approaches to visual odometry have gained a lot of attention. The advantage of learned methods is their potential robustness to lack of features, dynamic lighting conditions, motion blur, accurate camera calibration, which are hard to model by hand [31]. PoseNet [18] used Convolutional Neural Networks (CNNs) for 6-DoF pose regression from monocular images. The combination of CNNs and Long-Short Term Memory (LSTM) networks was reported in [7, 36], showing comparable results to traditional methods. Several approaches [43, 41, 42] used the view synthesis as unsupervisory signal to train and estimate both ego-motion and depth. Other DL-based methods can be found on learning representations for dense visual SLAM [3], general map [4], global pose [27], deep Localization and segmentation [35]. We study the contribution of multimodal data to robust deep localization in degraded scenarios.

**Multimodal Sensor fusion and Attention** Our proposed selective sensor fusion is related with the attention mechanisms, widely applied in neural machine translation [33], image caption generation [39], and video description [14]. Limited by the fixed-length vector in embedding space, these attention mechanisms compute a focus map to help the decoder, when generating a sequence of words. This is different from our design intention that the features selection works to fuse multimodal sensor fusion for visual inertial odometry, and cope with more complex error resources, and self-motion dynamics.

## 6. Conclusion

In this work, we presented a novel study of end-to-end sensor fusion for visual-inertial navigation. Two feature selection strategies are proposed: deterministic soft fusion, in which a soft mask is learned from the concatenated visual and inertial features, and a stochastic hard fusion, in which Gumbel-softmax resampling is used to learn a stochastic binary mask. Based on the extensive experiments, we also provided insightful interpretations of selective sensor fusion and investigate the influence of different modalities under different degradation and self-motion circumstances.



## References

- [1] C. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. 5
- [2] M. Bloesch, M. Burri, S. Omari, M. Hutter, and R. Siegwart. Iterated extended kalman filter visual-inertial odometry using direct photometric feedback. *The International Journal of Robotics Research*, 36(10):1053–1072, 2017. 5
- [3] M. Bloesch, J. Czarnowski, R. Clark, S. Leutenegger, and A. J. Davison. CodeSLAM Learning a Compact, Optimisable Representation for Dense Visual SLAM. In *CVPR*, 2018. 8
- [4] S. Brahmabhatt, J. Gu, K. Kim, J. Hays, and J. Kautz. Geometry-Aware Learning of Maps for Camera Localization. In *CVPR*, pages 2616–2625, 2018. 1, 8
- [5] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart. The euroc micro aerial vehicle datasets. *The International Journal of Robotics Research*, 2016. 5, 6, 7
- [6] C. Chen, C. X. Lu, A. Markham, and N. Trigoni. Ionet: Learning to cure the curse of drift in inertial odometry. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2018. 3
- [7] R. Clark, S. Wang, A. Markham, N. Trigoni, and H. Wen. VidLoc: A Deep Spatio-Temporal Model for 6-DoF Video-Clip Relocalization. In *CVPR*, 2017. 1, 8
- [8] F. Couzinie-Devy, J. Sun, K. Alahari, and J. Ponce. Learning to estimate and remove non-uniform image blur. In *CVPR*, pages 1075–1082, 2013. 6
- [9] C. R. Fetsch, A. H. Turner, G. C. DeAngelis, and D. E. Angelaki. Dynamic Reweighting of Visual and Vestibular Cues during Self-Motion Perception. *Journal of Neuroscience*, 29(49):15601–15612, 2009. 1
- [10] P. Fischer, E. Ilg, H. Philip, C. Hazrbas, P. V. D. Smagt, D. Cremers, and T. Brox. FlowNet: Learning Optical Flow with Convolutional Networks. In *International Conference on Computer Vision, ICCV*, 2015. 2
- [11] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza. On-manifold preintegration for real-time visual-inertial odometry. *IEEE Transactions on Robotics*, 33(1):1–21, 2017. 1, 3, 8
- [12] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 5, 6, 7
- [13] E. J. Gumbel. *Statistical theory of extreme values and some practical applications: a series of lectures*. U. S. Govt. Print. Office, 1954. 5
- [14] C. Hori, T. Hori, T. Y. Lee, Z. Zhang, B. Harsham, J. R. Hershey, T. K. Marks, and K. Sumi. Attention-Based Multimodal Fusion for Video Description. *Proceedings of the IEEE International Conference on Computer Vision*, 2017-October:4203–4212, 2017. 4, 8
- [15] J. S. Hu and M. Y. Chen. A sliding-window visual-IMU odometer based on tri-focal tensor geometry. In *ICRA*, pages 3963–3968. IEEE, 2014. 5, 7
- [16] E. Jang, S. Gu, and B. Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016. 4
- [17] E. S. Jones and S. Soatto. Visual-inertial navigation, mapping and localization: A scalable real-time causal approach. *The International Journal of Robotics Research*, 30(4):407–430, 2011. 1, 8
- [18] A. Kendall, M. Grimes, and R. Cipolla. PoseNet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE international conference on computer vision*, pages 2938–2946, 2015. 1, 2, 8
- [19] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale. Keyframe-based visual-inertial odometry using non-linear optimization. *The International Journal of Robotics Research*, 34(3):314–334, 2015. 1, 7, 8
- [20] M. Li and A. I. Mourikis. High-precision, Consistent EKF-based Visual-Inertial Odometry. *The International Journal of Robotics Research*, 32(6):690–711, 2013. 1, 5, 6
- [21] Y. Ling, L. Bao, Z. Jie, F. Zhu, Z. Li, S. Tang, Y. Liu, W. Liu, and T. Zhang. Modeling Varying Camera-IMU Time Offset in Optimization-Based Visual-Inertial Odometry. In *The European Conference on Computer Vision (ECCV)*, 2018. 1, 6
- [22] C. J. Maddison, A. Mnih, and Y. W. Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016. 4
- [23] C. J. Maddison, D. Tarlow, and T. Minka. A\* Sampling. In *NIPS*, pages 1–9, 2014. 5
- [24] A. Mnih and K. Gregor. Neural variational inference and learning in belief networks. *arXiv preprint arXiv:1402.0030*, 2014. 4
- [25] A. I. Mourikis and S. I. Roumeliotis. A multi-state constraint Kalman filter for vision-aided inertial navigation. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 3565–3572, 2007. 5
- [26] N. Naser, El-Sheimy; Haiying, Hou; Xiaojii. Analysis and Modeling of Inertial Sensors Using Allan Variance. *IEEE Transactions on Instrumentation and Measurement*, 57(JANUARY):684–694, 2008. 1, 6
- [27] E. Parisotto, D. S. Chaplot, J. Zhang, and R. Salakhutdinov. Global Pose Estimation with an Attention-based Recurrent Network. In *CVPR*, 2018. 8
- [28] B. Pfrommer, N. Sanket, K. Daniilidis, and J. Cleveland. PenncoSyvio: A challenging visual inertial odometry benchmark. In *2017 IEEE International Conference on Robotics and Automation, ICRA 2017, Singapore, Singapore, May 29 - June 3, 2017*, pages 3847–3854, 2017. 5, 6, 7
- [29] T. Qin, P. Li, and S. Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4):1004–1020, Aug 2018. 1, 7
- [30] H. W. A. M. N. T. Ronald Clark, Sen Wang. Vinet: Visual-inertial odometry as a sequence-to-sequence learning problem. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence. AAAI*, 2017. 1, 6, 8
- [31] N. Sünderhauf, O. Brock, W. Scheirer, R. Hadsell, D. Fox, J. Leitner, B. Upcroft, P. Abbeel, W. Burgard, M. Milford, and P. Corke. The limits and potentials of deep learning for robotics. *International Journal of Robotics Research*, 37(4-5):405–420, 2018. 8

- [32] P. Tanskanen, T. Naegeli, M. Pollefeys, and O. Hilliges. Semi-direct ekf-based monocular visual-inertial odometry. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 6073–6078. IEEE, 2015. 8
- [33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention Is All You Need. In *NIPS*, 2017. 4, 8
- [34] L. von Stumberg, V. Usenko, and D. Cremers. Direct sparse visual-inertial odometry using dynamic marginalization, 2018. 8
- [35] P. Wang, R. Yang, B. Cao, W. Xu, and Y. Lin. DeLS-3D: Deep Localization and Segmentation with a 3D Semantic Map. In *CVPR*, 2018. 8
- [36] S. Wang, R. Clark, H. Wen, and N. Trigoni. Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks. *International Conference on Robotics and Automation*, 2017. 6, 8
- [37] T. C. Wang, A. A. Efros, and R. Ramamoorthi. Occlusion-aware depth estimation using light-field cameras. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3487–3495, 2015. 6
- [38] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992. 4
- [39] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *ICML*, 2015. 4, 8
- [40] N. Yang, R. Wang, X. Gao, and D. Cremers. Challenges in Monocular Visual Odometry: Photometric Calibration, Motion Bias and Rolling Shutter Effect. *IEEE ROBOTICS AND AUTOMATION LETTERS*, pages 1–8, 2018. 1
- [41] Z. Yin and J. Shi. GeoNet: Unsupervised Learning of Dense Depth, Optical Flow and Camera Pose. In *CVPR*, 2018. 2, 8
- [42] H. Zhan, R. Garg, C. S. Weerasekera, K. Li, H. Agarwal, and I. Reid. Unsupervised Learning of Monocular Depth Estimation and Visual Odometry with Deep Feature Reconstruction. In *CVPR*, pages 340–349, 2018. 2, 8
- [43] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, volume 2, page 7, 2017. 2, 8

# Supplementary Material

## Abstract

*In this supplementary document we provide additional implementation details, including the detailed network architecture, and data preparation/training details. We also provide additional results and discussion on predicted global trajectories.*

## 1. Network Architecture

The implementation details for the architecture that was used in our experiments are reported in Table 1. We use the following abbreviations:

**Conv**: convolution, **FC**: fully-connected, **activ.**: activation function,  $\oplus$ : concatenation;  $\odot$ : element-wise multiplication;  $B$ : batch size.

The visual encoder follows the encoder structure of the FlowNetS architecture [3]. We initialize the visual encoder weights with the weights of a model that was pre-trained on the FlyingChairs dataset<sup>1</sup>, since training from scratch would require larger amounts of data compared to our dataset sizes. For this reason, training the encoder from scratch experimentally gave worse results.

Bi-directional LSTMs were used for the inertial encoder (as in [2]) as well as in the recurrent part of the pose regressor. Bi-directional RNNs provide higher data efficiency when dealing with small datasets. The LSTM layers include dropout regularization on the recurrent connections.

## 2. Dataset Preprocessing Details

Few publicly available datasets provide camera images, high-frequency IMU and ground-truth for training visual inertial odometry (i.e., inertial data is missing in the Oxford Robotcar Dataset [5] and 7-Scenes Dataset [9]; ground-truth for the full trajectories is not available in TUM VIO [8]). Therefore, we use the KITTI [4], EuRoC [1], and PennCOSYVIO [7] datasets for training and evaluation.

**KITTI** High-frequency inertial data (100Hz) is only available in the raw unsynced data packages. Hence, we manually synchronize inertial data and images according to

their timestamps. We used Sequences 00, 01, 02, 04, 06, 08, 09 for training and tested the network on Sequences 05, 07, and 10, excluding sequence 03 as the corresponding raw file is unavailable. Correspondences between the KITTI Odometry Number, the raw data file names and the corresponding number of sequences are reported in Table 2. The images are resized to  $512 \times 256$ .

**EuRoC** For the EuRoC MAV dataset [1] we use grayscale video images from CAM1 at 20fps of the VI-Sensor and the IMU measurements from the same sensors at 200Hz. The data is tightly synchronized. Doubling the sensor rates compared to the ones in KITTI helps to deal with the faster, jerky MAV movements. As with KITTI, images are resized to  $512 \times 256$ . We train on all sequences, minus *MH\_04\_difficult*, which is used for testing.

**PennCOSYVIO** A number of sensors are included in the PennCOSYVIO dataset [7]. For our experiment we select the video images from the Tango Bottom camera, running at 30fps, and inertial measurements from the VI-Sensor, running at 200Hz. Images are subsampled to 10Hz, and IMU measurements are subsampled to 100Hz, similarly to KITTI. Images are cropped and resized to  $512 \times 256$ . We train on sequences *as*, *bf*, *bs* and test on sequence *af*. Small time-synchronization errors between the Tango camera and the VI-Sensor are likely present in the data, leading to worse results in all scenarios.

## 3. Evaluation of global trajectories on KITTI

Figure 1 shows the global RMSE position errors on the three test KITTI trajectories (Seq 05, Seq 07, Seq 10) as a function of travelled distance, for both the normal dataset and the fully degraded dataset (visual degradation + sensor degradation). We compare the two VO and vanilla VIO baselines with the proposed soft and hard fusion strategies. It can be noticed how, while at start VIO performs as well as soft and hard fusion, on average, over time the proposed selective fusion strategies outperform the vanilla fusion, since the increased robustness reduces error accumulation. This is particularly visible in the most challenging Seq 05. As expected, a VO approach heavily underperforms in presence of large amounts of angular rotations (Seq 05, Seq 07).

Another interesting result is how VO performs slightly better in presence of IMU degradation and camera-IMU

<sup>1</sup><https://lmb.informatik.uni-freiburg.de/resources/datasets/FlyingChairs.en.html>

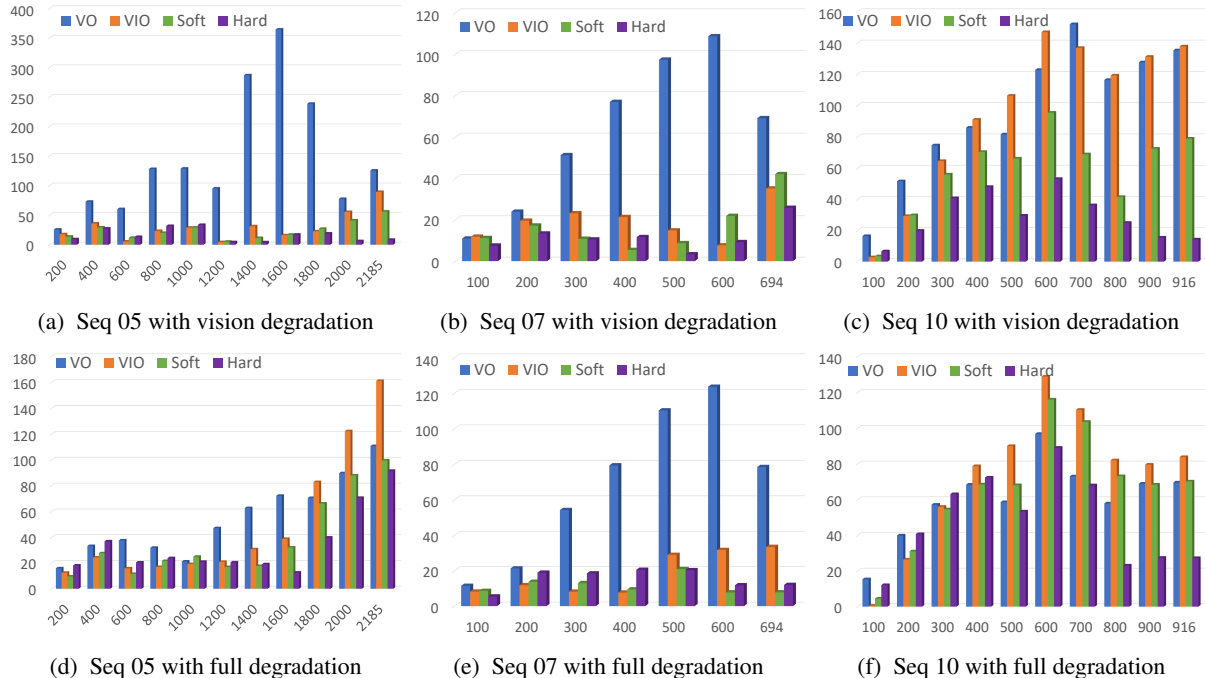


Figure 1: Global position errors (in meters, Y axis) on the KITTI dataset, over travelled distance (in meters, X axis).

synchronization (Figure 1, bottom row). That shows how a vanilla VIO fusion is unable to deal with these issues, to the point of underperforming compared to vision-only approaches. This result further corroborates the fact that in deep learning-based approaches explicitly learning the belief on the different components makes the estimation more robust, while stacking sensors without a sensible fusion strategy can lead to *catastrophic fusion*, similarly to traditional approaches. Catastrophic fusion happens when the single components of the system before fusion significantly outperform the overall system after fusion [6].

#### 4. Supplementary Video

The supplementary video shows an evaluation of soft fusion and hard fusion on Seq 05 of the KITTI dataset, with vision degradation (10% occlusion, 10% blur+noise, 10% missing data), compared with two deep VO and VIO baselines. The degraded images and corresponding soft/hard masks are shown in the top-left and bottom-left respectively. The trajectories from the four methods are shown on the right side.

#### References

- [1] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart. The euroc micro aerial vehicle datasets. *The International Journal of Robotics Research*, 2016. 1
- [2] C. Chen, C. X. Lu, A. Markham, and N. Trigoni. Ionet: Learning to cure the curse of drift in inertial odometry. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2018. 1
- [3] P. Fischer, E. Ilg, H. Philip, C. Hazrbas, P. V. D. Smagt, D. Cremers, and T. Brox. FlowNet: Learning Optical Flow with Convolutional Networks. In *International Conference on Computer Vision, ICCV*, 2015. 1
- [4] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 1
- [5] W. Maddern, G. Pascoe, C. Linegar, and P. Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *The International Journal of Robotics Research (IJRR)*, 36(1):3–15, 2016. 1
- [6] J. R. Movellan and P. Mineiro. Robust sensor fusion: Analysis and application to audio visual speech recognition. *Machine Learning*, 32(2):85–100, 1998. 2
- [7] B. Pfrommer, N. Sanket, K. Daniilidis, and J. Cleveland. Penncosyvio: A challenging visual inertial odometry benchmark. In *2017 IEEE International Conference on Robotics and Automation, ICRA 2017, Singapore, Singapore, May 29 - June 3, 2017*, pages 3847–3854, 2017. 1
- [8] D. Schubert, T. Goll, N. Demmel, V. Usenko, J. Stückler, and D. Cremers. The TUM VI Benchmark for Evaluating Visual-Inertial Odometry. In *ICRA*, 2018. 1
- [9] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon. Scene coordinate regression forests for camera relocalization in RGB-D images. In *CVPR*, pages 2930–2937, 2013. 1

### Visual Encoder

---

[ input ]	Two stacked images: $B \times 512 \times 256 \times 6$
[ layer 1 ]	Conv. $7^2$ , Stride $2^2$ , Padding 3, LeakyReLU activ.
[ layer 2 ]	Conv. $5^2$ , Stride $2^2$ , Padding 2, LeakyReLU activ.
[ layer 3 ]	Conv. $5^2$ , Stride $2^2$ , Padding 2, LeakyReLU activ.
[ layer 3.1 ]	Conv. $3^2$ , Stride $1^2$ , Padding 1
[ layer 4 ]	Conv. $3^2$ , Stride $2^2$ , Padding 2, LeakyReLU activ.
[ layer 4.1 ]	Conv. $3^2$ , Stride $1^2$ , Padding 1
[ layer 5 ]	Conv. $3^2$ , Stride $2^2$ , Padding 2, LeakyReLU activ.
[ layer 5.1 ]	Conv. $3^2$ , Stride $1^2$ , Padding 1
[ layer 6 ]	Conv. $3^2$ , Stride $2^2$ , Padding 1
[ layer 6 output ]	$B \times 4 \times 8 \times 1024$
[ layer 7 ]	FC 256
[ output ]	$B \times 256$

### Inertial Encoder

---

[ input ]	IMU sequence: $B \times 10 \times 6$
[ layer 1 ]	FC 128
[ layer 2 ]	B-LSTM, 2-layers, hidden size 128, Dropout 0.2
[ output ]	$B \times 256$

### Direct Fusion Module

---

[ input ]	$B \times (256 \oplus 256)$
[ output ]	$B \times 512$

### Soft Fusion Module

---

[ input ]	$B \times (256 \oplus 256)$
[ layer 1 ]	FC 512 (input)
[ layer 2 ]	$[(\text{input}) \odot (\text{layer 1})]$
[ output ]	$B \times 512$

### Hard Fusion Module

---

[ input ]	$B \times (256 \oplus 256)$
[ layer 1 ]	FC 1024, Sigmoid activ. (input)
[ layer 2 ]	Gumbel-Softmax sampling, $512 \times 2$
[ layer 2 ]	$[(\text{input}) \odot (\text{layer 2})]$
[ output ]	$B \times 512$

### Pose Regressor

---

[ input ]	$B \times 512$
[ layer 1 ]	B-LSTM, 2-layers, hidden size 512, Dropout 0.2
[ layer 2 ]	Dropout 0.2
[ layer 3.1 ]	FC 3 (layer 2)
[ layer 3.2 ]	FC 3 (layer 2)
[ layer 4 ]	$(\text{layer 3.1}) \oplus (\text{layer 3.2})$
[ output ]	$B \times 6$

Table 1: Implementation details for the proposed architecture.  $\oplus$  denotes a concatenation operation;  $\odot$  indicates an element-wise product between two tensors.

Table 2: Correspondences between the KITTI Odometry Number, the raw data file names and the corresponding number of sequences

Odometry Nr.	Raw Data Filename	Number of Sequences
00	2011_10_03_drive_0027	4390
01	2011_10_03_drive_0042	1173
02	2011_10_03_drive_0034	4485
03	2011_09_26_drive_0067	Not available
04	2011_09_30_drive_0016	284
05	2011_09_30_drive_0018	2750
06	2011_09_30_drive_0020	1091
07	2011_09_30_drive_0027	1111
08	2011_09_30_drive_0028	5149
09	2011_09_30_drive_0033	1599
10	2011_09_30_drive_0034	1215