VINet: Visual-Inertial Odometry as a Sequence-to-Sequence Learning Problem

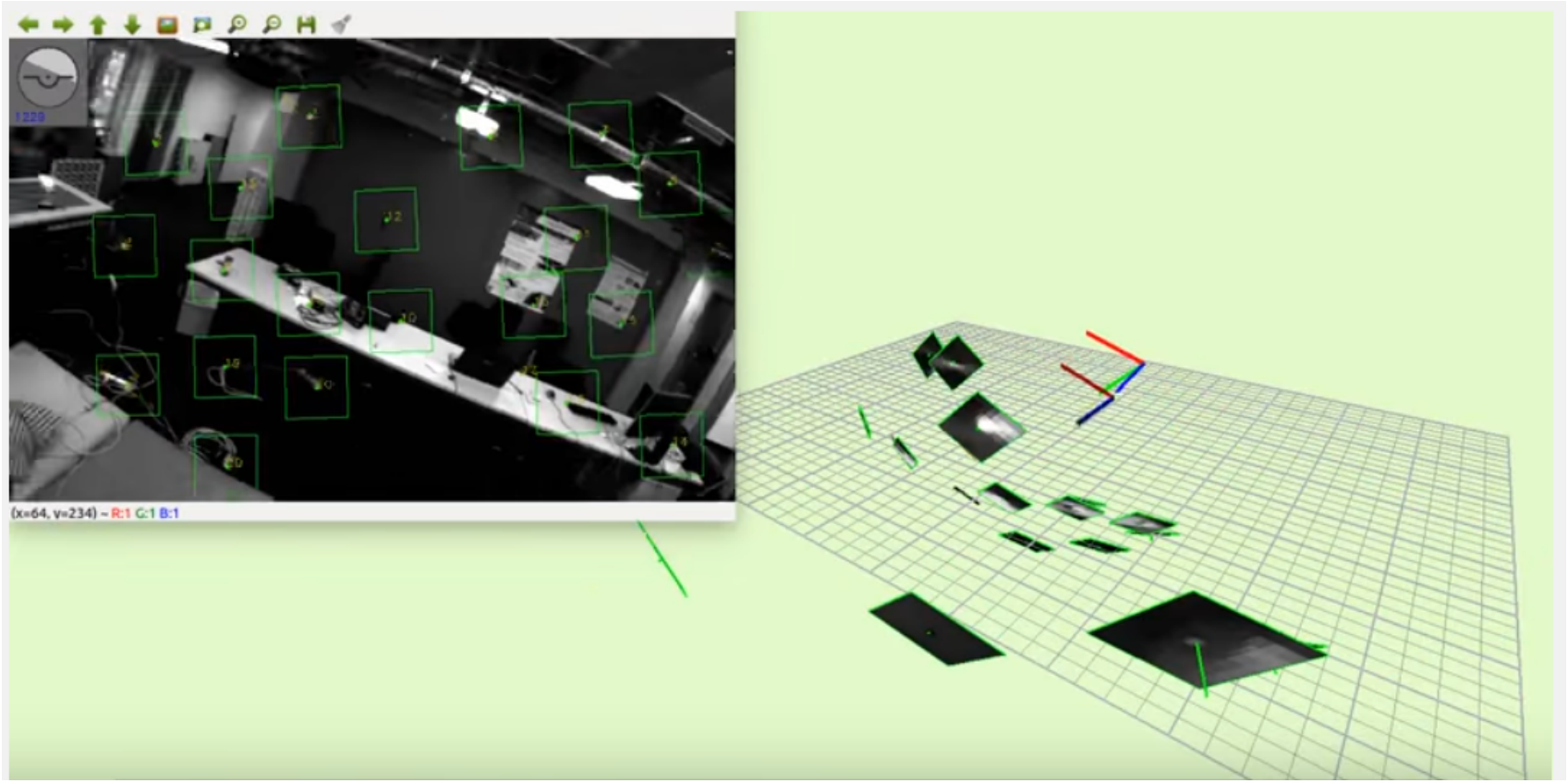
Presented by: Justin Gorgen Yen-ting Chen Hao-en Sung Haifeng Huang

University of California, San Diego

January 12, 2018

Original paper by: Ronald Clark[1] , Sen Wang[1] , Hongkai Wen , Andrew Markham[1] and Niki Trigoni[1], arXiv: 1701.08376

[1]Department of Computer Science,University of Oxford, United Kingdom
[1]Department of Computer Science,University of Warwick, United Kingdom

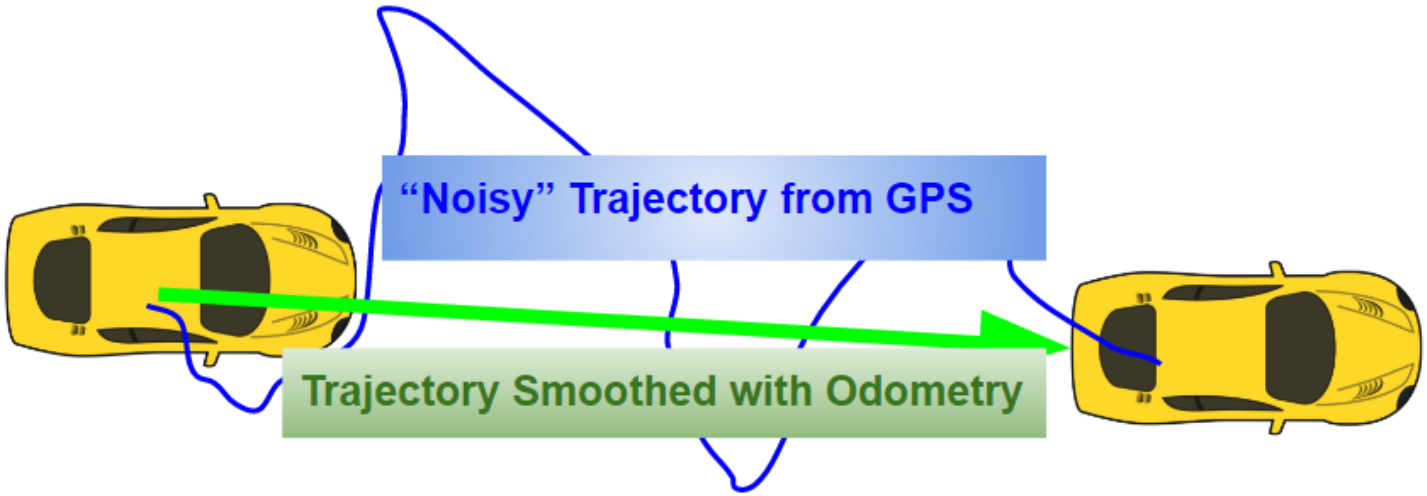


Visual-Inertial Odometry Example

Odometry

Odometry is the use of data from sensors to estimate change in position (Δp) over time.

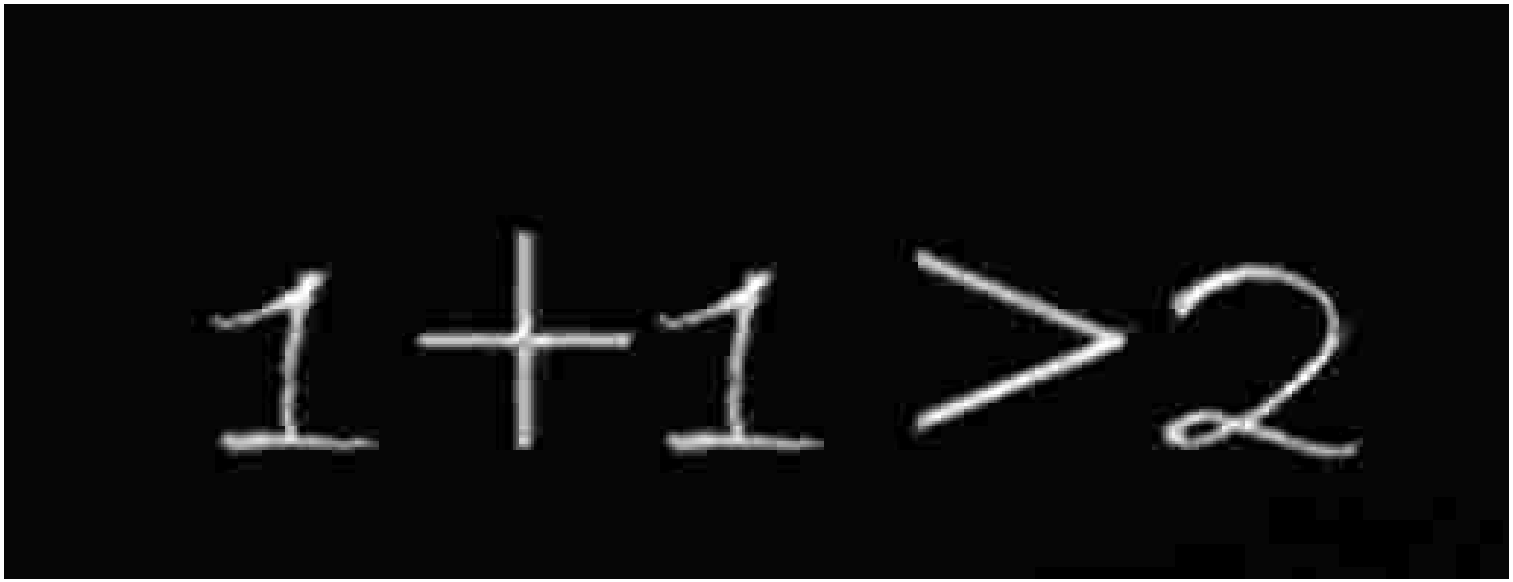
- The odometer in a car counts wheel rotations to estimate distance driven.
- Odometry is used in navigation systems to smooth out GPS.



Visual-Inertial Odometry

Visual-Inertial Odometry is the use of an opto-electronic device (i.e. a camera) and an inertial measurement unit (IMU) to calculate Δp.

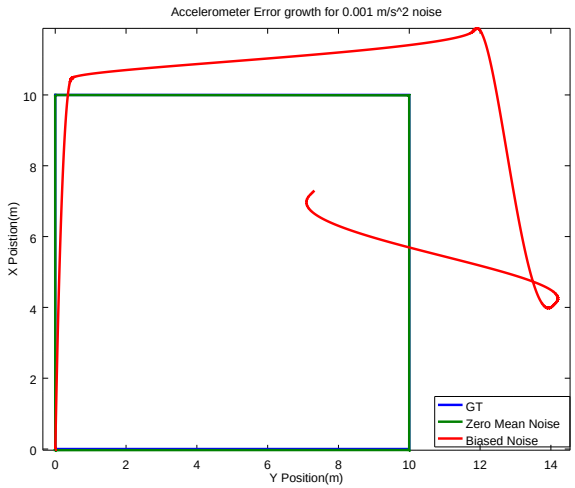
- Cameras and IMUs have complementary weaknesses as odometry sensors.
- Fusing Camera and IMU data should allow for better odometry than either sensor is capable of independently.



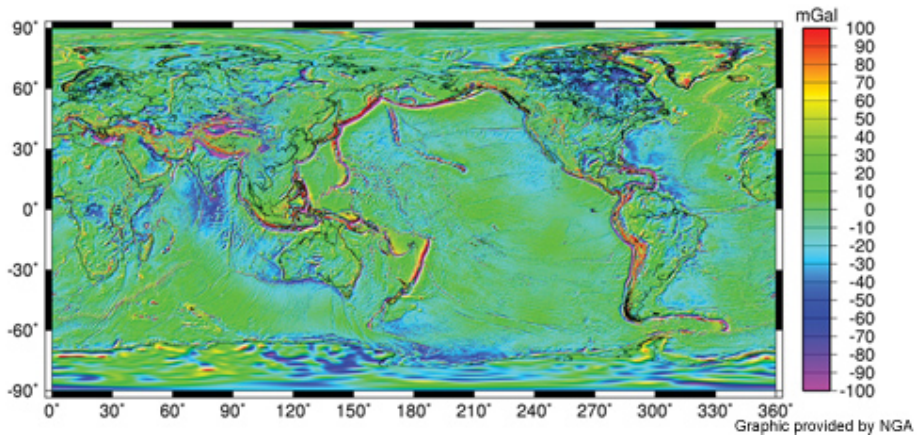
Why can't we just use an IMU?

- **An IMU** consists of gyroscopes and accelerometers that can measure the rotations and accelerations of an object in 3D.
- **Error growth** is catastrophic if you just integrate accelerometer to calculate velocity and integrate velocity to calculate Δp.
  - A small constant error in acceleration leads to a linearly growing error in velocity, and a **quadratic** error in position.
  - Similarly, a small constant gyroscope error in roll-rate or pitch-rate leads to a **cubic** error in position.

IMU problems



(a) IMU errors accumulate quickly



(b) Gravity anomaly maps are needed if you really want to use just an IMU

Why can't we just use a camera?

- **A monocular camera** has no sense of scale.
  - Current efforts to learn scale based on visible objects (i.e. cars and estimate) have limited success.
- **Stereo cameras** give a sense of scale, but are vulnerable to calibration issues.

monocular

stereo

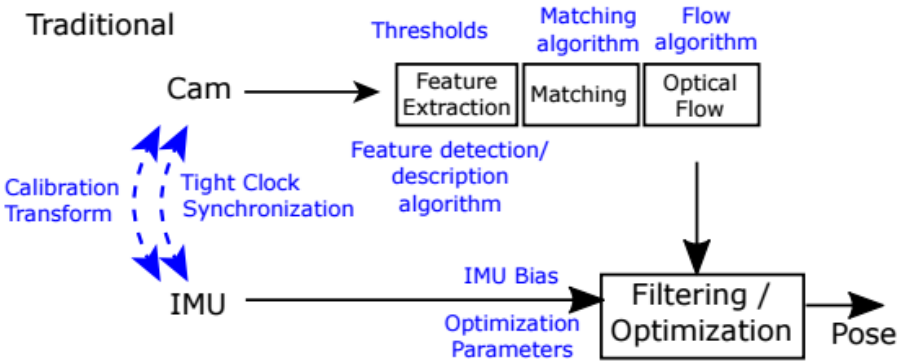
Common approaches to Visual-Inertial Odometry

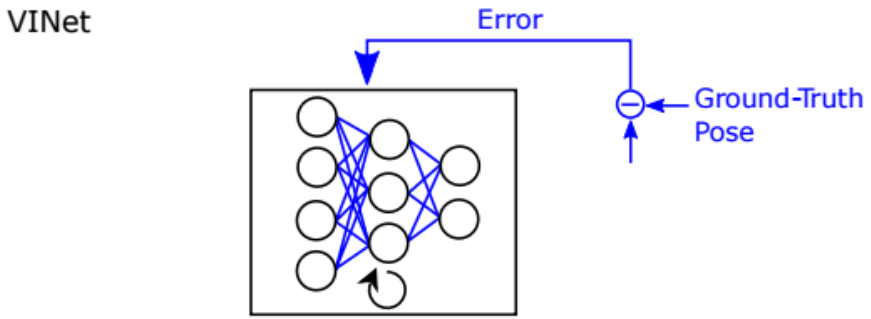
- **Fusing IMU and Camera data** is traditionally accomplished through the use of an EKF or particle filter.
- **SLAM** (Simultaneous Localization and Mapping) is required in order to constrain error growth (state of the art is 0.2% error per distance traveled).
- **SLAM-based approaches** involve:
  - Identifying salient features in each frame of a video sequence.
  - Matching features between each frame (ostensibly requires N^2 comparisons for matching N features, but various heuristics have been used to greatly speed this process).
  - Storing a database of features.
  - Calculating a camera pose from the visible features.
  - Calculating a correction for the feature locations to improve future position estimates.
- All these processes require hand-tuning based on the expected motion profile of the camera system.

Motivation

- Developing a SLAM or Visual Odometry architecture is mathematically easy.
  - "Just" do bundle adjustment to solve for pose.
  - Identifying, matching, and managing feature points is hard.
  - Managing non-linear error cases is hard.
- SLAM and VO systems are hand-tuned by picking algorithms that minimize errors on training data, then tested on a new data set.
- Motivation: machine-learning should be able to minimize our error for us, by back-propagating visual odometry error to weights on input pixels and memory states.

Comparison of traditional framework and VINet



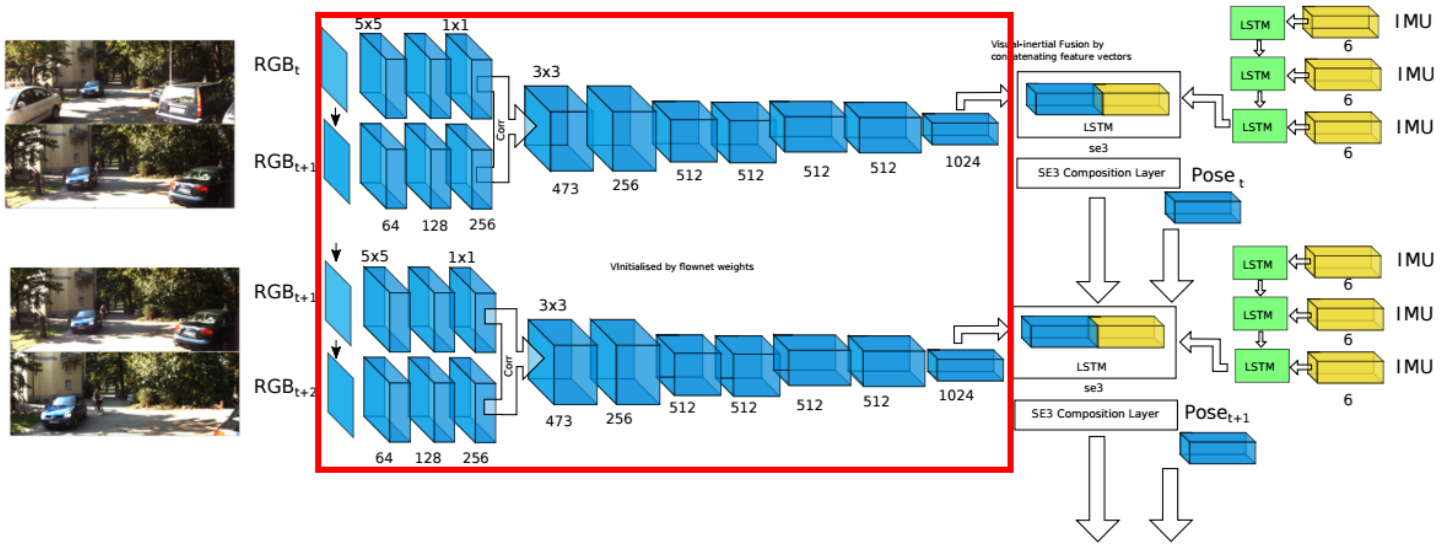


- Feature extraction and matching is slow
- Subject to scale drift and scale ambiguity
- Suffer from calibration error
- Need tight clock synchronization

- Prediction is fast
- Can solve the problem of scale drift and scale ambiguity
- Can be robust to calibration error
- Performs well with time-synchronization error

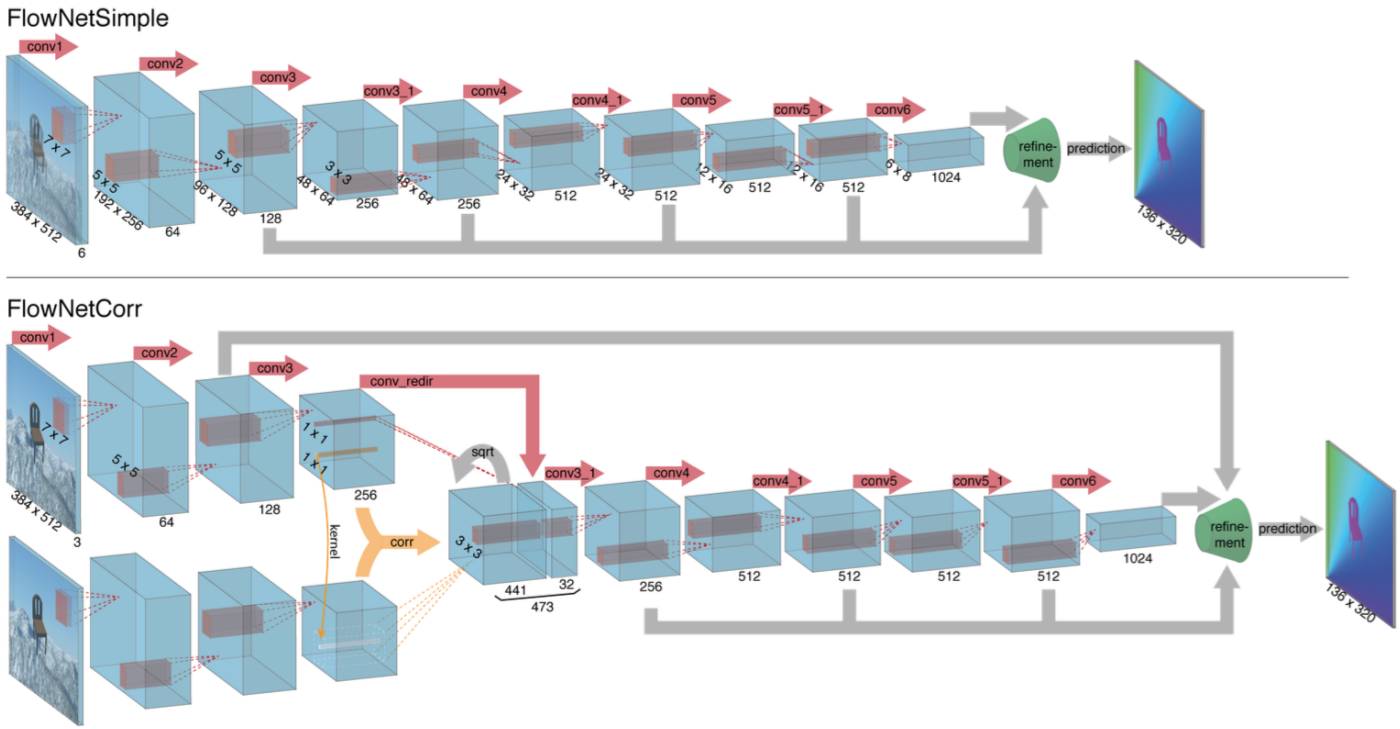
Optical Flow Weight Initialization

- They initialized the convolutional neural network using Imagenet at first, but they found that the convergence was slow and the performance is not very well, so they used the Flownet which is trained to predict optical flow from RGB images to initialize the network up to the Conv6 layer.



FlowNet in Framework

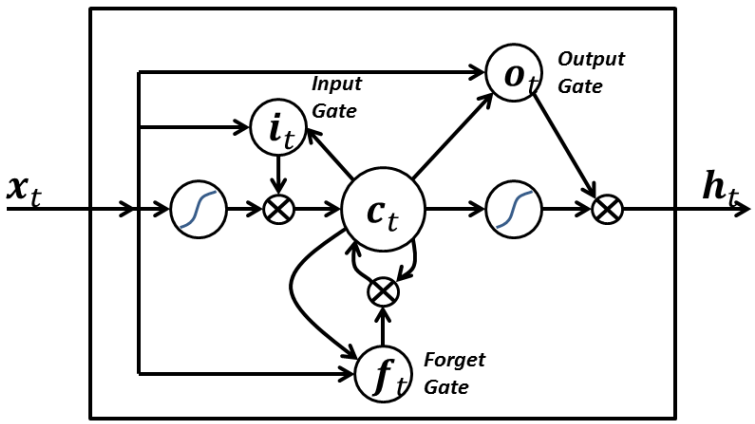
Convolutional Neural Network (CNN) - FlowNet



FlowNet Framework

Long Short-term Memory (LSTM)

- Why use LSTM?
  - RNN has the disadvantage that using standard training techniques, they are unable to learn to store and operate on long-term trends in the input and thus do not provide much benefit over standard feed-forward networks.
  - For this reason, the Long Short-Term Memory (LSTM) architecture was introduced to allow RNN to learn longer-term trends (Hochreiter and Schmidhuber 1997).
  - This is accomplished through the inclusion of gating cells which allow the network to selectively store and forget memories.



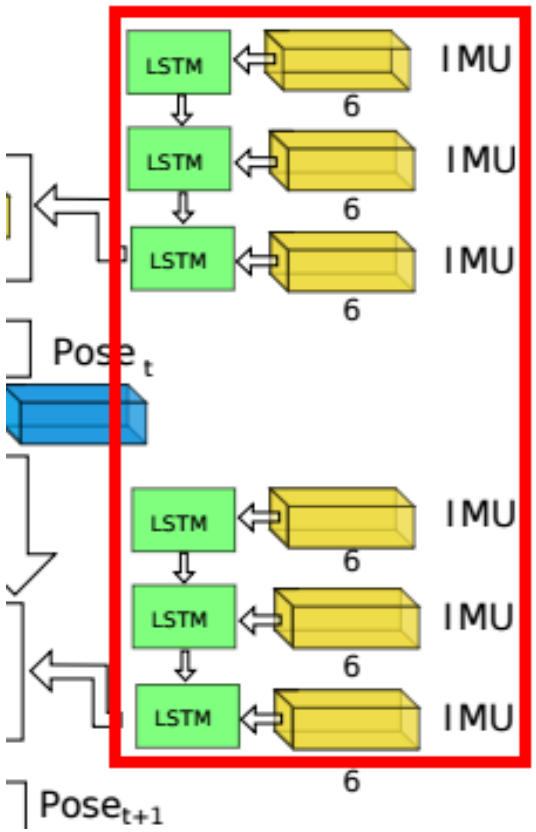
LSTM structure

$$i_t = \sigma(\mathbf{W}_{xi}x_t + \mathbf{W}_{hi}h_{t-1} + \mathbf{W}_{ci}c_{t-1} + b_i) \quad (1)$$
$$f_t = \sigma(\mathbf{W}_{xf}x_t + \mathbf{W}_{hf}h_{t-1} + \mathbf{W}_{cf}c_{t-1} + b_f) \quad (2)$$
$$z_t = \tanh(\mathbf{W}_{xc}x_t + \mathbf{W}_{hc}h_{t-1} + b_c) \quad (3)$$
$$c_t = f_t \odot c_{t-1} + i_t \odot z_t \quad (4)$$
$$o_t = \sigma(\mathbf{W}_{xo}x_t + \mathbf{W}_{ho}h_{t-1} + \mathbf{W}_{co}c_t + b_o) \quad (5)$$
$$h_t = o_t \odot \tanh(c_t) \quad (6)$$

LSTM equations

Multi-rate LSTM

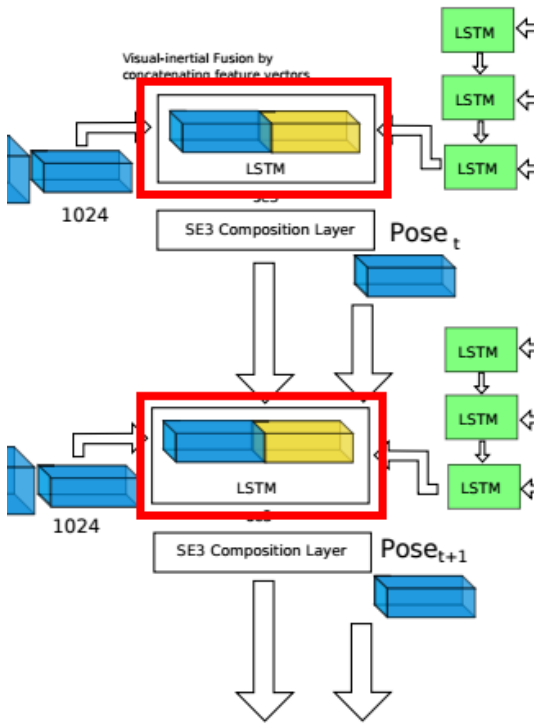
- Because the IMU data(100 Hz) arrives 10 times faster than the visual data(10 Hz), the authors processed the IMU data using a small LSTM at the IMU rate, and fed that result to the core LSTM.



IMU LSTM in Framework

Core LSTM

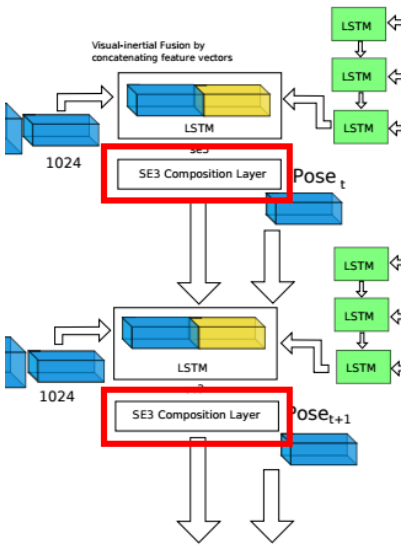
- Before being fed into core LSTM, the result vector of IMU data and flattened feature vectors of CNN is concatenated.
- The output of the core LSTM is 6-dimensional vector, which contains 3-dimensional angle-axis representation and 3-dimensional translation vector



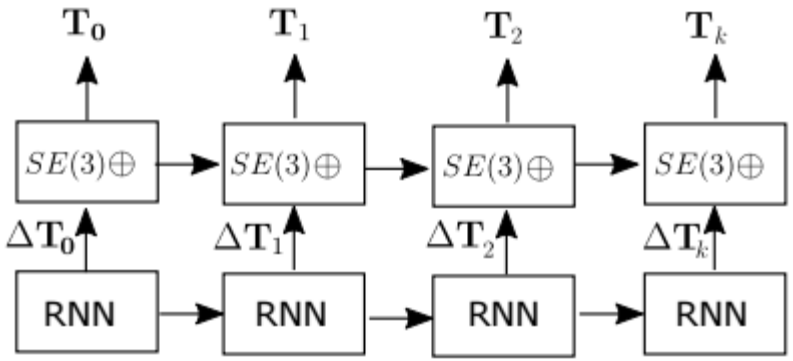
Core LSTM in Framework

SE(3) Concatenation of Transformations

- The CNN-RNN thus performs the mapping from the input data to the lie algebra se(3) of the current time step. An exponential map is used to convert these to the special euclidean group SE(3) and right multiply it to the cumulated SE(3) of the last time step to get the cumulated sum of current time step.



SE3 Layer in Framework



SE(3) composition layer

SE(3) Concatenation of Transformations

- The pose of a camera relative to an initial starting point is conventionally represented as an element of the special Euclidean group SE(3) of transformations.

$$T = \left\{ \begin{pmatrix} R & T \\ 0 & 1 \end{pmatrix} | R \in SO(3), T \in R^3 \right\} \tag{1}$$

- However, the Lie Algebra se(3) of SE(3), representing the instantaneous transformation

$$\frac{\xi}{dt} = \left\{ \begin{pmatrix} [\omega]_\times & v \\ 0 & 0 \end{pmatrix} | \omega \in so(3), v \in R^3 \right\}, \tag{2}$$

can be described by components which are not subject to orthogonality constraints. Conversion between se(3) and SE(3) is then easily accomplished using the exponential map.

$$exp : se(3) \rightarrow SE(3) \tag{3}$$

Loss function as printed in the paper:

- Frame-to-frame pose (se(3)):

$$\mathcal{L}_{se(3)} = \alpha \sum \|\omega - \hat{\omega}\| + \beta \|v - \hat{v}\|$$

$$B = \omega \otimes \omega = \frac{1}{2}(R + I)$$

The diagonal terms of B are the squares of the elements of ω and the signs (up to sign ambiguity) can be determined from the signs of the off-axis terms of B.

- Full pose (SE(3)):

$$\mathcal{L}_{SE(3)} = \alpha \sum \|\boldsymbol{q} - \hat{\boldsymbol{q}}\| + \beta \left\| T - \hat{T} \right\|$$

Because **q** is homogeneous, we will constrain $\|\mathbf{q}\| = 1$. Presenters' Note: this is a **poor** loss function to use for attitude:

  - Quaternions provide a double-cover of SO3 (i.e. **q** and -**q** represent the same rotation)
  - Quaternion errors in the axis and angle of rotation are not orthogonal
  - As written in the paper, α and β are not independent

Better Loss function

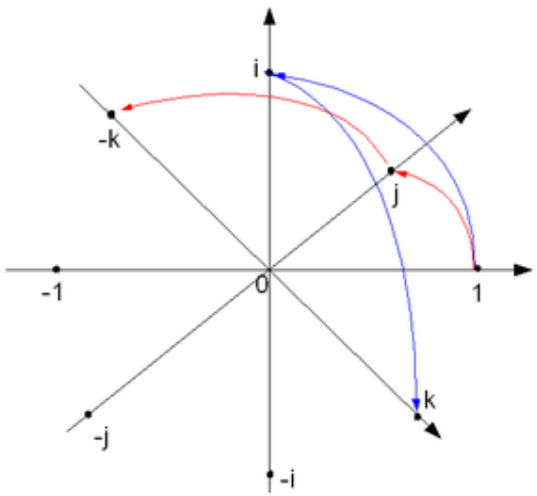
The presenters suggest a better loss function for SE3:

$$\mathcal{L}_{SE(3)} = \sum \alpha \left\| \hat{\omega} \right\| + \beta \left\| T - \hat{T} \right\|$$

Here, $\hat{\omega}$ is defined to be the minimum se3 representation of the SO3 rotation $\boldsymbol{q}^{-1}\hat{\boldsymbol{q}}$.

Quaternion - 3D Rotation Representation

- Advantages
  - More compact than the matrix representation and less susceptible to round-off errors
  - The quaternion elements vary continuously over the unit sphere in R^4 as the orientation changes, avoiding discontinuity jumps (inherent to three-dimensional parameterizations)
  - Expression of the rotation matrix in terms of quaternion parameters involves no trigonometric functions
  - It is simple to combine two individual rotations represented as quaternions using a quaternion product



Training Procedure

- Joint training algorithm:

Algorithm 1 Joint training of $se(3)$ and $SE(3)$ loss

while $i \leq n_{iter}$ **do**

$$w^{1:n} = w^{1:n} - \lambda_1 \frac{\partial \mathcal{L}_{SE(3)}(w^l, x_t)}{\partial w^l}$$

$$w^{1:j} = w^{1:j} - \lambda_2 \frac{\partial \mathcal{L}_{se(3)}(w^l, x_t)}{\partial w^l}$$

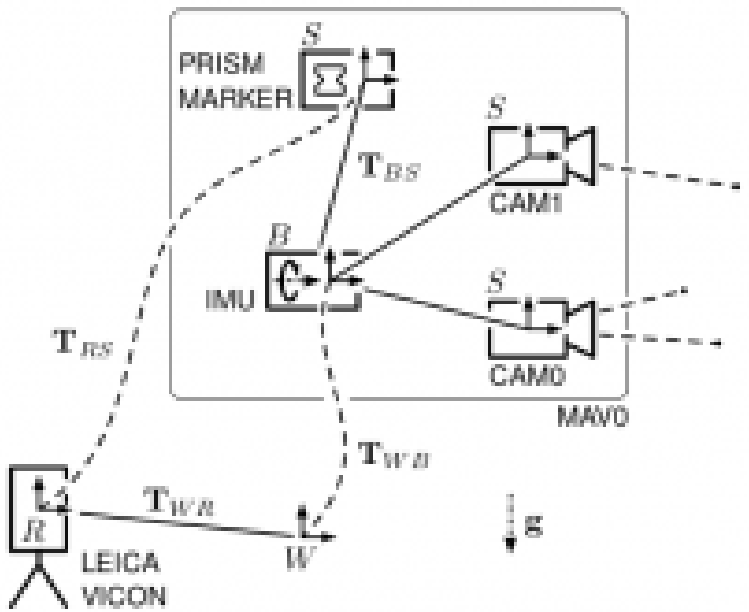
end while

- First train the SE(3) concatenation layer using full pose loss function, treat ω and v as input, **q** and **T**.
- Then train all the weights using frame-to-frame loss function, treat images and IMU data as input, ω and v as output.
- Start with a high relative learning rate for the se(3) loss with $\frac{\lambda_2}{\lambda_1} \approx 100$.
- During the later epochs, fine-tune the concatenated pose estimation with $\frac{\lambda_2}{\lambda_1} \approx 0.01$.

Robustness Experiment Design

- Compare reconstructed trajectory and numerical results between:
  - Proposed VINet model
  - Optimization-based OK-VIS (Leutenegger et al. 2015)
- Examine the robustness against camera calibration errors.
  - A scaler is chosen as magnitude
  - An angle is randomly chosen ($\Delta R_{SC} \sim vMF(\cdot|\mu, \kappa)$)
- Run robustness experiments on MAV dataset.

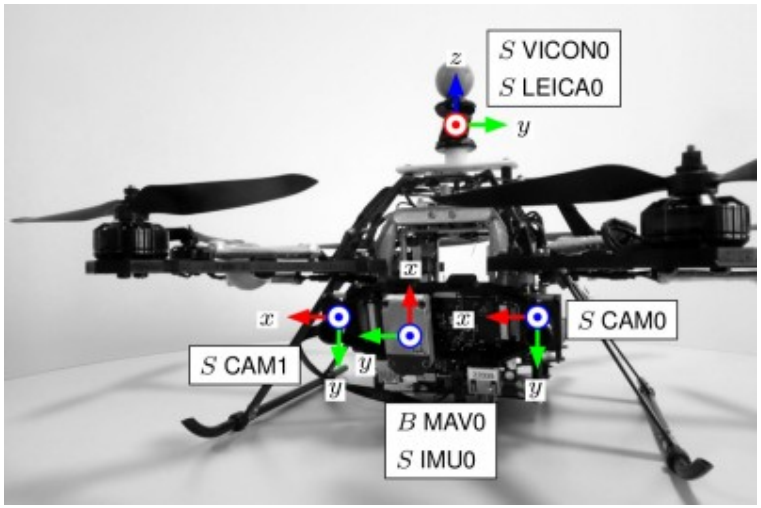
monocular

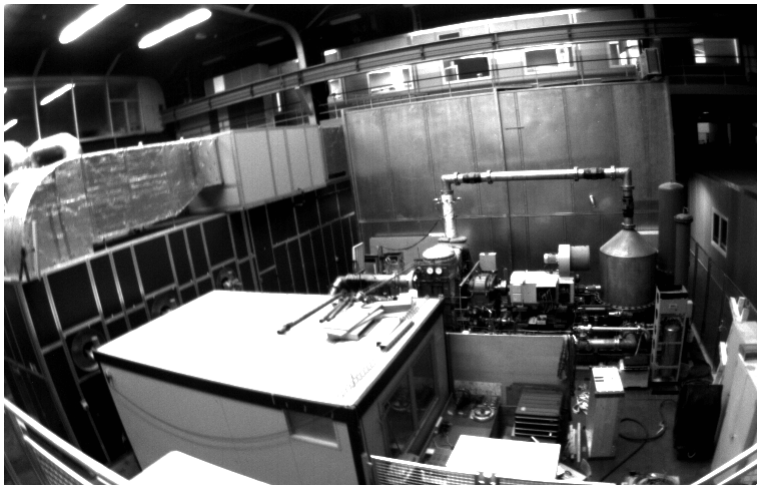


stereo

Introduction to MAV Dataset

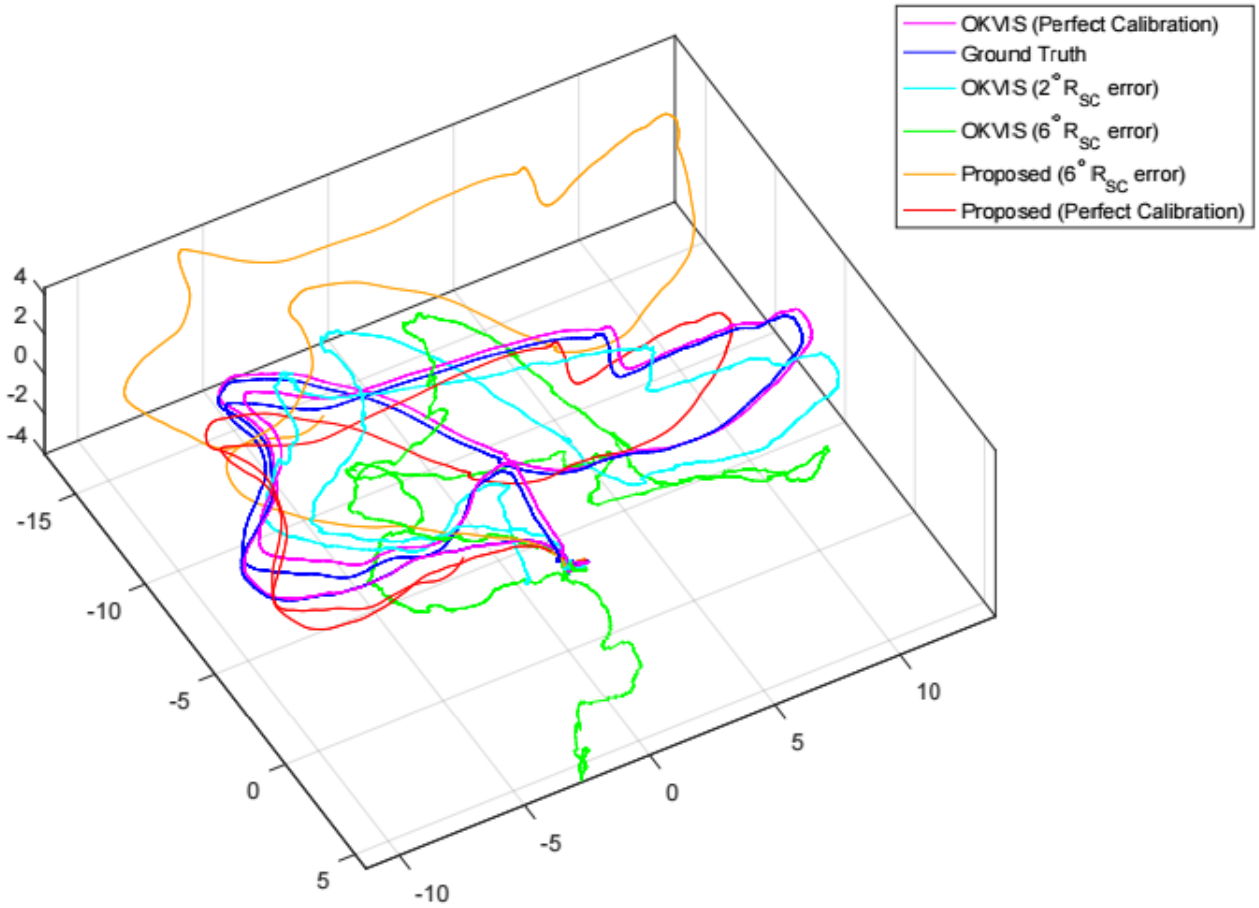
- EuRoC micro-aerial-vehicle (MAV) dataset (Burri et al. 2016)
- It is an indoor trajectory dataset.
- Provided information includes:
  - Images are captured by shutter camera (20 Hz)
  - Accelerations and angular rates are captured by IMU (200 Hz)
  - 6-D ground truth poses are captured by VICON (100 Hz)





Micro Aerial Vehicle (MAV) and Example

MAV Trajectory Results



MAV Reconstructed Trajectory for Robustness Test

MAV Numerical Results

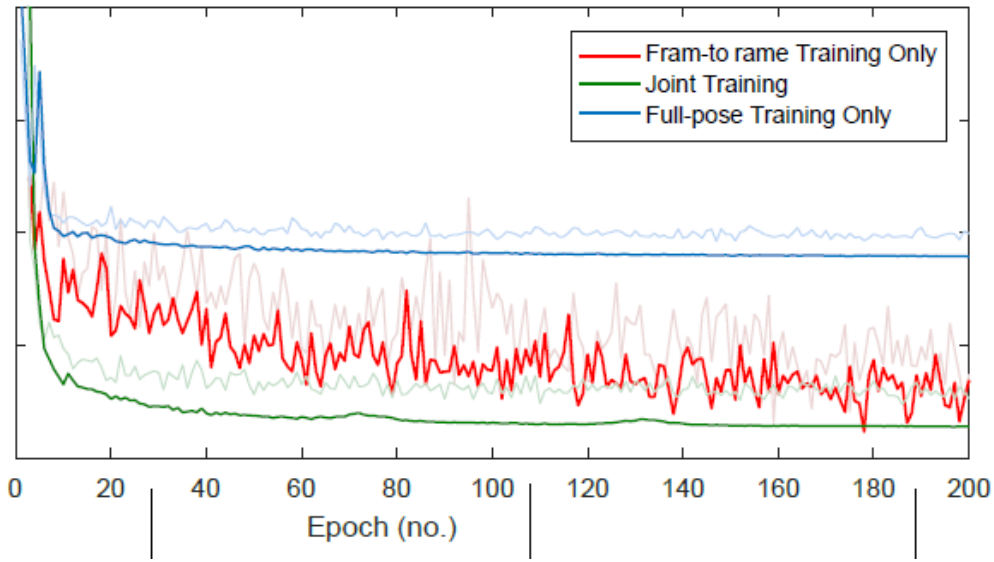
	0°	5°	10°	15°
VI Net (no-aug)	0.1751	0.8023	1.94	3.0671
VI Net (w/ aug)	0.1842	0.1951	0.2218	0.5178
OK-VIS	0.1644	0.7916	1.9138	FAILS

MAV Numerical Results for Robustness Test

- VINet (w/ aug) means that VINet also considers some further artificially mis-calibrated training data (called augmentation), which makes it more general and more robust against calibration errors in testing stage.
- According to MAV dataset, there is no synchronization issue between camera and IMU; while this issue exists when considering ground truth.

Results of Different Loss Function

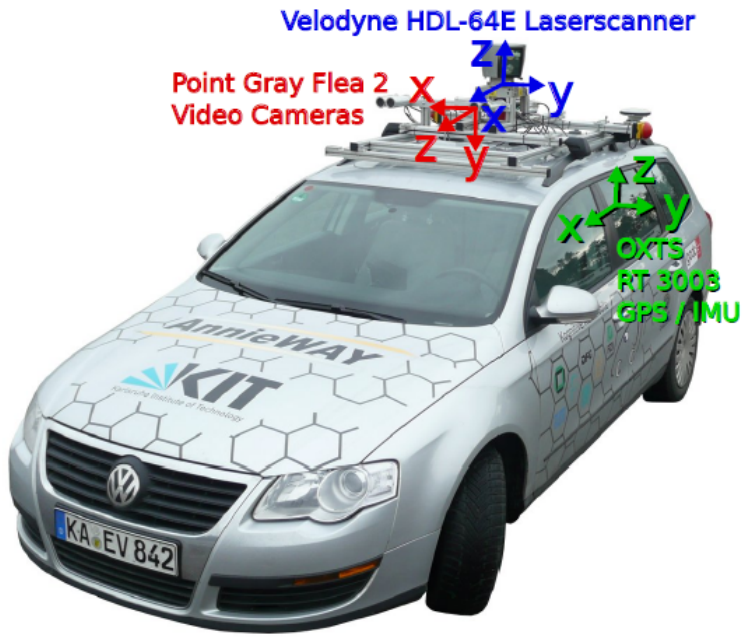
- Mentioned in previous slides, there are three different methods to train the model using different objective function:
  - Frame-to-frame displacement
  - SE(3) pose
  - Joint training



Model Performance with different loss functions

Introduction to KITTI Dataset

- KITTI odometry benchmark (Geiger, Lenz, and Urtasun 2012; Geiger et al. 2013).
- Data are recorded from a moving-platform while driving around Karlsruhe, Germany.



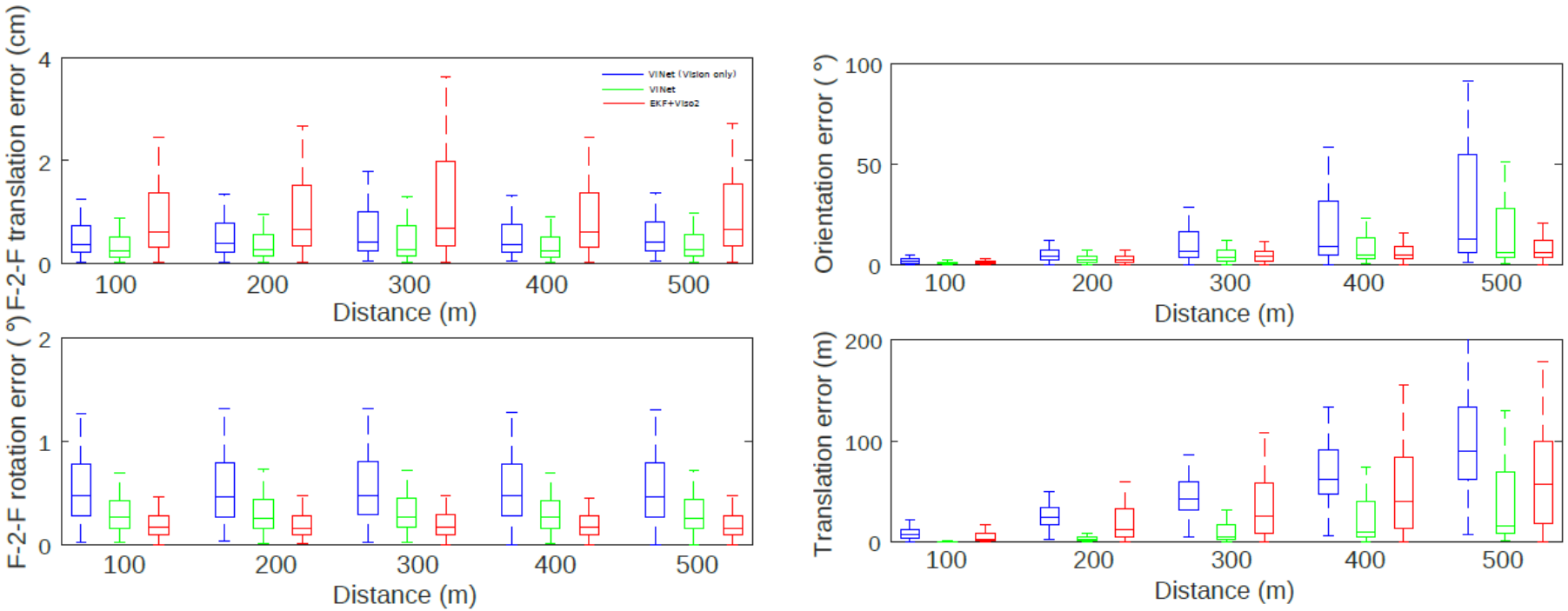
KITTI recording platform

Introduction to KITTI Dataset

- 11 sequence of frames recorded in outdoor scenario including city, residential, campus, and road.
- Sampling rate: frames(10Hz) and IMU data(100Hz)

KITTI data example

Outdoor Trajectory Performance



Translation and orientation on the KITTI dataset

- VINet outperforms the network using visual data alone.
- VINet outperforms VISO2 in translation error, but in terms of orientation, VISO2 is better.

Conclusion

	Traditional Method	VINet
Robustness		better
Translation		better
Orientation	slightly better	

- VINet performs on-par with traditional way, while VINet do not require efforts of hand-tuning

Q&A