Convolution Neural Network based Sensors for Mobile Robot Relocalization

Harsh Sinha, Jay Patrikar, Eeshan Dhekane, Gaurav Pandey, Mangal Kothari

IIT Kanpur

TEAM IGVC
INDIAN INSTITUTE OF TECHNOLOGY KANPUR

Intelligent Guidance & Control Laboratory

Introduction

Motivation

Results

Methodology

Experiments

Implementation Details

IIT
Kanpur

Introduction

Motivation

Results

Methodology

Experiments

Implementation Details

IIT
Kanpur

Introduction

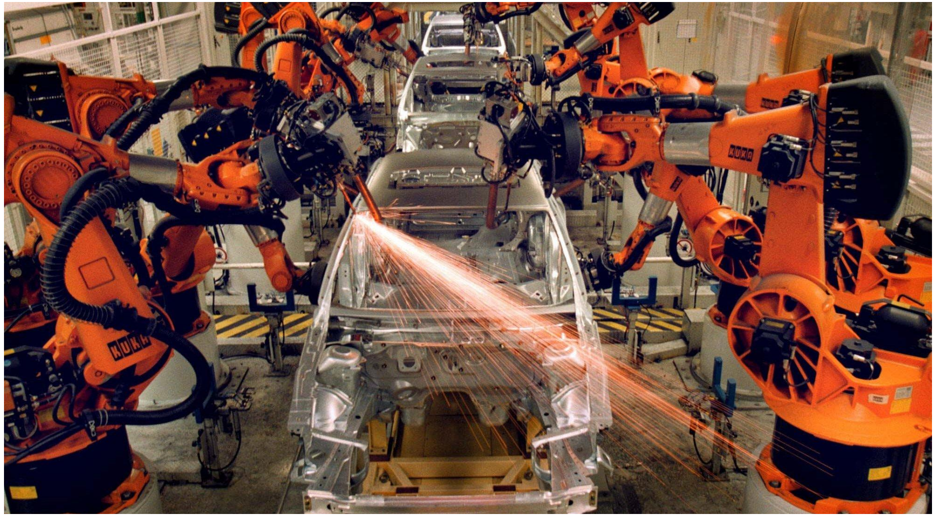
We have proposed here a real-time shallow CNN based architecture which combines low-cost sensors of a mobile robot with information from images of a single monocular camera using an Extended Kalman Filter to perform accurate robot relocalization.

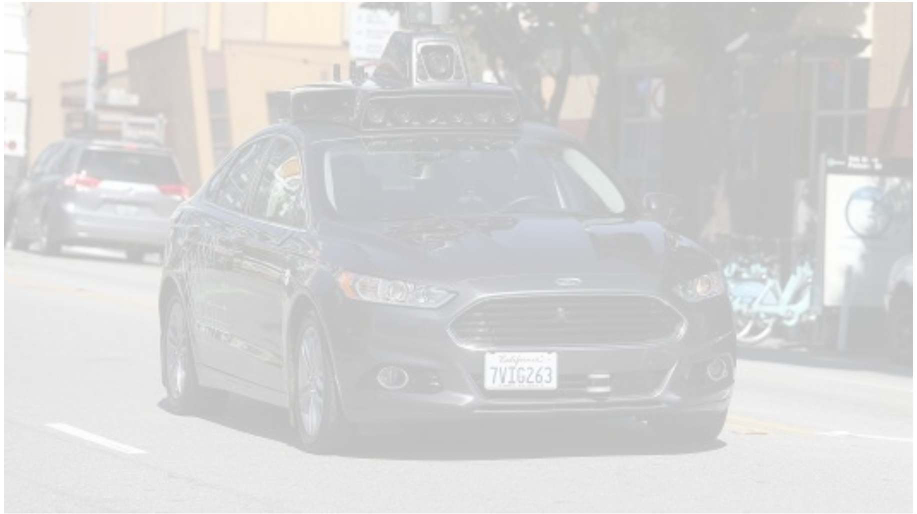
Introduction

We have proposed here a **real-time shallow CNN** based architecture which combines **low-cost sensors** of a **mobile robot** with information from images of a **single monocular camera** using an Extended Kalman Filter to perform accurate robot **relocalization**.

Mobile Robots ?

Mobile Robots

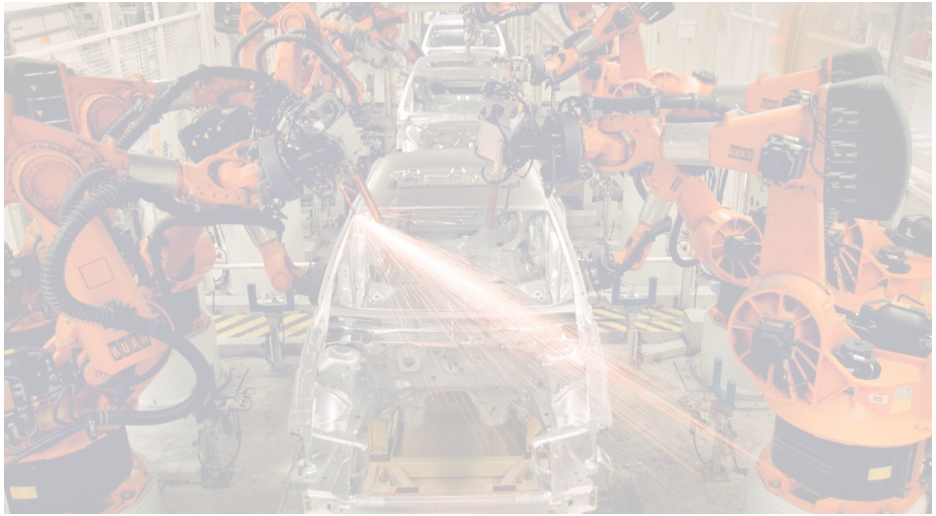




Robots have gone from being exclusively **Fixed** on factory floors ...

* No Copyright infringement intended, Images taken from https://images.techhive.com and https://fm.cnbc.com

Mobile Robots



… to **Autonomously Roaming** around the world.

IIT
Kanpur

* No Copyright infringement intended, Images taken from https://images.techhive.com and https://fm.cnbc.com

What’s with Relocalization ?

Relocalization Vs Localization

Localization: The ability of a mobile entity to infer its position in a predefined frame of reference. For instance the location of your car on Google Maps.

Relocalization Vs Localization

Localization: The ability of a mobile entity to infer its position in a predefined frame of reference. For instance the location of your car on Google Maps.

ReLocalization: The ability to localize **again** in an environment after using the information from a localization done earlier.

Relocalization Vs Localization

Localization: The ability of a mobile entity to infer its position in a predefined frame of reference. For instance the location of your car on Google Maps.

ReLocalization: The ability to localize **again** in an environment after using the information from a localization done earlier.

In the method proposed we use the information from the localization done to train our network.

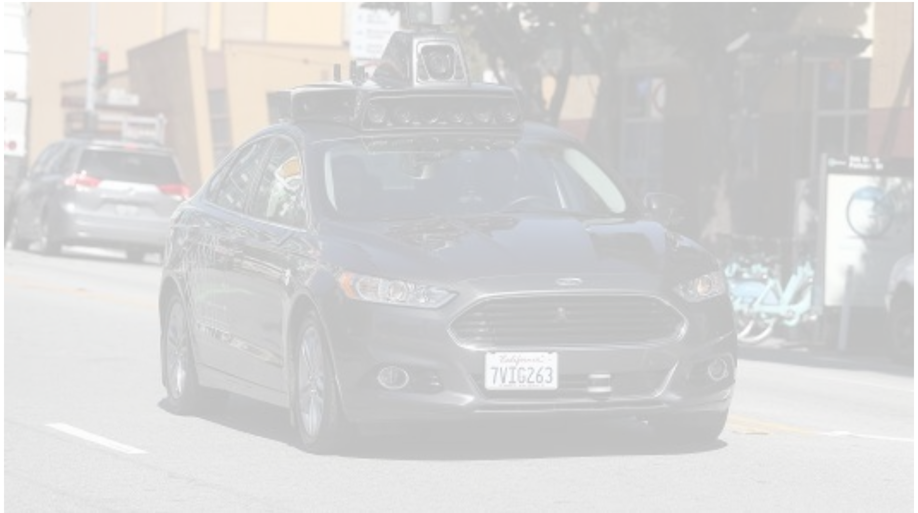
Why do we need cheap, low-power ?

Cheap Low-Power Sensors

The sensor suit a car can afford to be Costly, Heavy and Power Hungry ….

* No Copyright infringement intended, Images taken from https://fm.cnbc.com

Cheap Low-Power Sensors



The sensor suit a car can afford to be Costly, Heavy and Power Hungry ….

…. Not small mobile robots meant to function in a variety of environments.

IIT Kanpur

* No Copyright infringement intended, Images taken from https://fm.cnbc.com

Introduction

Motivation

Results

Methodology

Experiments

Implementation Details

IIT
Kanpur

Introduction

Motivation

Results

Methodology

Experiments

Implementation Details

Why do we need such a system ?

Motivation

- Majority of mobile robots employed in factories and laboratories are **moderate to low speed** vehicles with **moderate weight carrying capacities**, usually with **small batteries** onboard.

Motivation

- Majority of mobile robots employed in factories and laboratories are **moderate to low speed** vehicles with **moderate weight carrying capacities**, usually with **small batteries** onboard.
- Power hungry sensors would limit the duration of operation.

Motivation

- Environments where mobile robots work are usually **small** (of the order of ~$10^{1\text{-}2}$ m)

Motivation

- Environments where mobile robots work are usually **small** (of the order of ~$10^{1\text{-}2}$ m)
- Deep Network based architectures like **PoseNet** can model large environments though **require better hardware**, which is difficult to install on small mobile robots.

Motivation

- Environments where mobile robots work are usually small (of the order of ~10^{1-2} m)
- Deep Network based architectures like **PoseNet** can model large environments though **require better hardware**, which is difficult to install on small mobile robots.
- In our method we propose a shallower 8 layers network.

Introduction

Motivation

Results

Methodology

Experiments

Implementation Details

Introduction

Motivation

Results

Methodology

Experiments

Implementation Details

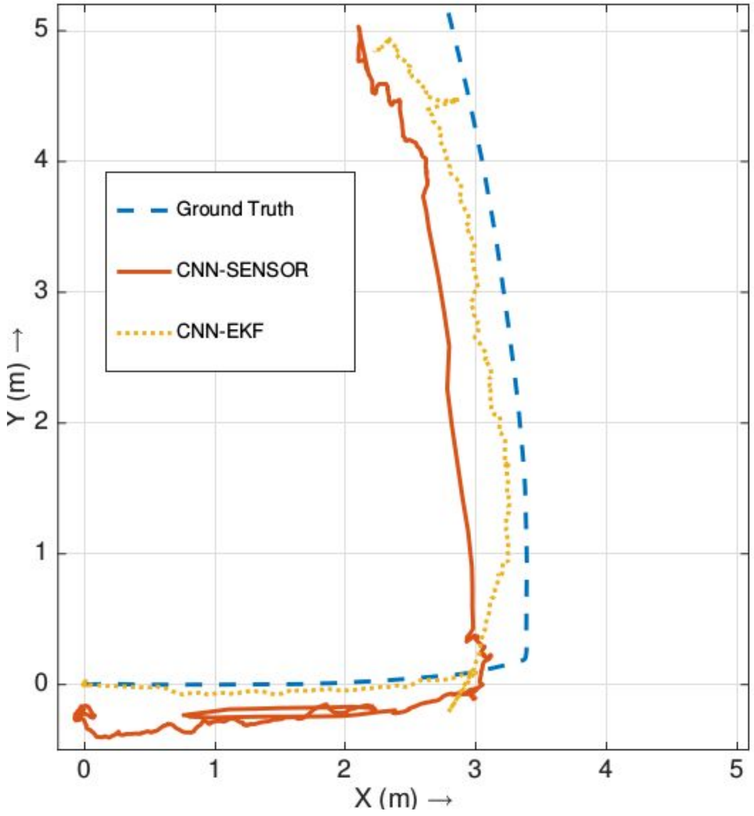
IIT
Kanpur

Does it work then?

IIT
Kanpur

Results

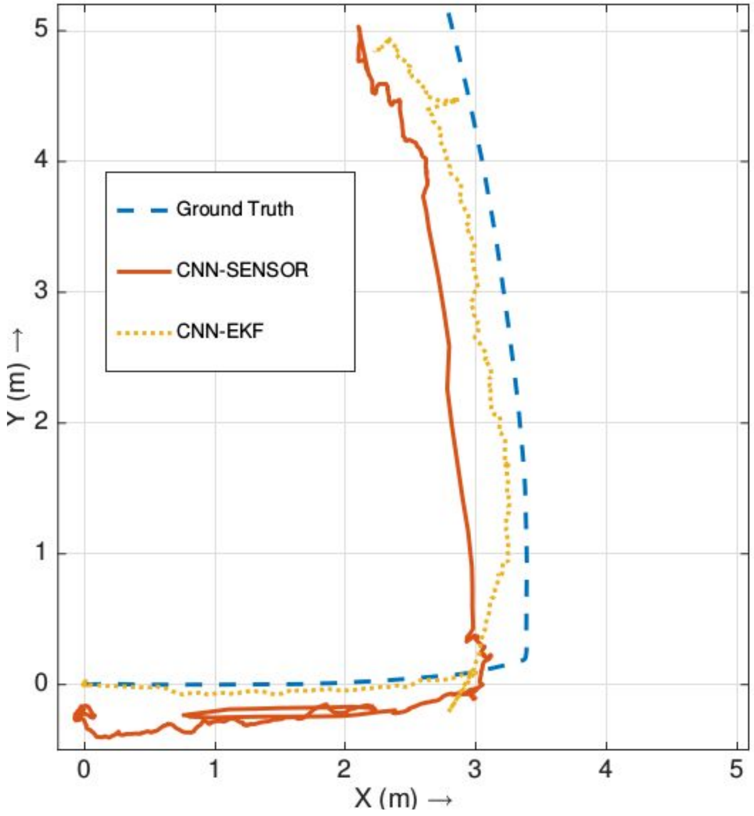
Yes, we tested the system on indoor and outdoor environment.



Results

Yes, we tested the system on indoor and outdoor environment.

Does it ever **fail**?

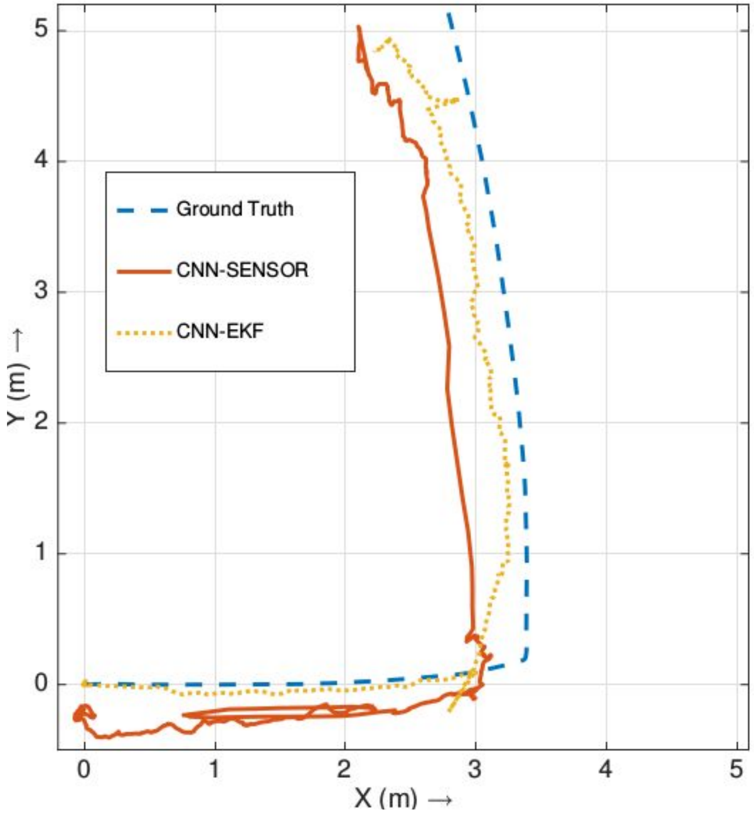


IIT
Kanpur

Results

Yes, we tested the system on indoor and outdoor environment.

Does it ever **fail**?
Yes, we investigated the failure conditions. They are included later.



Introduction

Motivation

Results

Methodology

Experiments

Implementation Details

IIT
Kanpur

- Introduction
- Motivation
- Results
- **Methodology**
- Experiments
- Implementation Details

IIT
Kanpur

How did we do it ?

Methodology

We proposed the following:

- **CNN Sensor**
- **CNN-EKF architecture.**

CNN Sensors

With CNN for real world application, we had to keep the constraints posed by mobile robots in mind:

- **Real-Time operation** on mobile robots: Frequency ~15 Hz
- **Computation constraints**: small memory *etc.*

CNN Sensors

With CNN for real world application, we had to keep the constraints posed by mobile robots in mind:

- **Real-Time operation** on mobile robots: Frequency ~15 Hz
- **Computation constraints**: small memory *etc.*

Thus, we use a modified Convolutional Neural Network similar to **AlexNet** as a position sensor, which we term **CNN Sensor**.

CNN Sensors

Our network has 8 layers:

- First **5 `conv`** layers with **ReLU** nonlinear activation, Layers **`conv1`**, **`conv2`** and **`conv5`** are followed by **pooling** layers.
- Next 2 layers are **`fully connected`** with 4096 neurons each.
- Last layer is again **`fully connected`** and provides 2 outputs, $\mathbf{x}_{\mathbf{CNN}}$, $\mathbf{y}_{\mathbf{CNN}}$

CNN Sensors

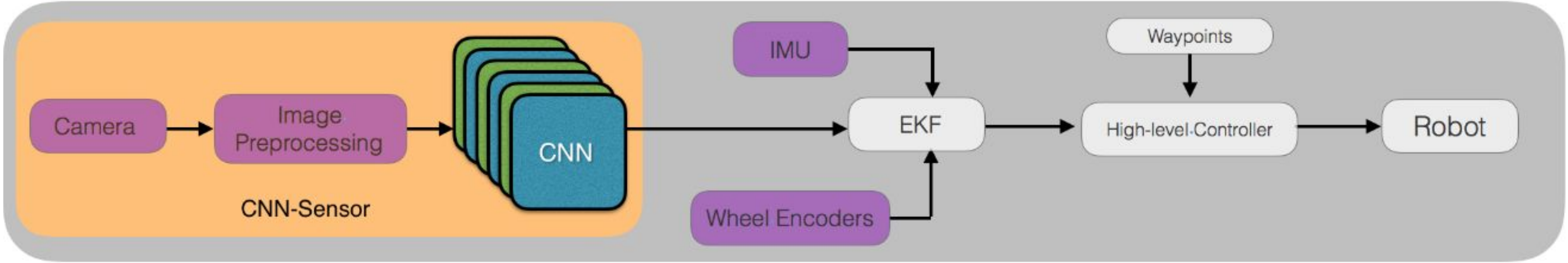
Our network has 8 layers:

- First **5 `conv`** layers with **ReLU** nonlinear activation, Layers **`conv1`**, **`conv2`** and **`conv5`** are followed by **pooling** layers.
- Next 2 layers are **`fully connected`** with 4096 neurons each.
- Last layer is again **`fully connected`** and provides 2 outputs, $\mathbf{x}_{\mathbf{CNN}}$, $\mathbf{y}_{\mathbf{CNN}}$

Loss Function : $\mathcal{L}_{\mathbf{x}} = \|\hat{\mathbf{x}} - \mathbf{x}\|_2$, where **x** is the regression value and $\hat{\mathbf{x}}$ is the ground truth.

CNN-EKF

We fuse the information from cheap sensors found on mobile robots with our CNN sensors using an Extended Kalman Filter in the CNN-EKF architecture as shown below:



Introduction

Motivation

Results

Methodology

Experiments

Implementation Details

Introduction

Motivation

Results

Methodology

Experiments

Implementation Details

Introduction

Motivation

Results

Methodology

Experiments

Indoor Experimentation

Outdoor Experimentation

IIT
Kanpur

Introduction

Motivation

Results

Methodology

Experiments

Indoor Experimentation

Outdoor Experimentation

Indoor Experimentation

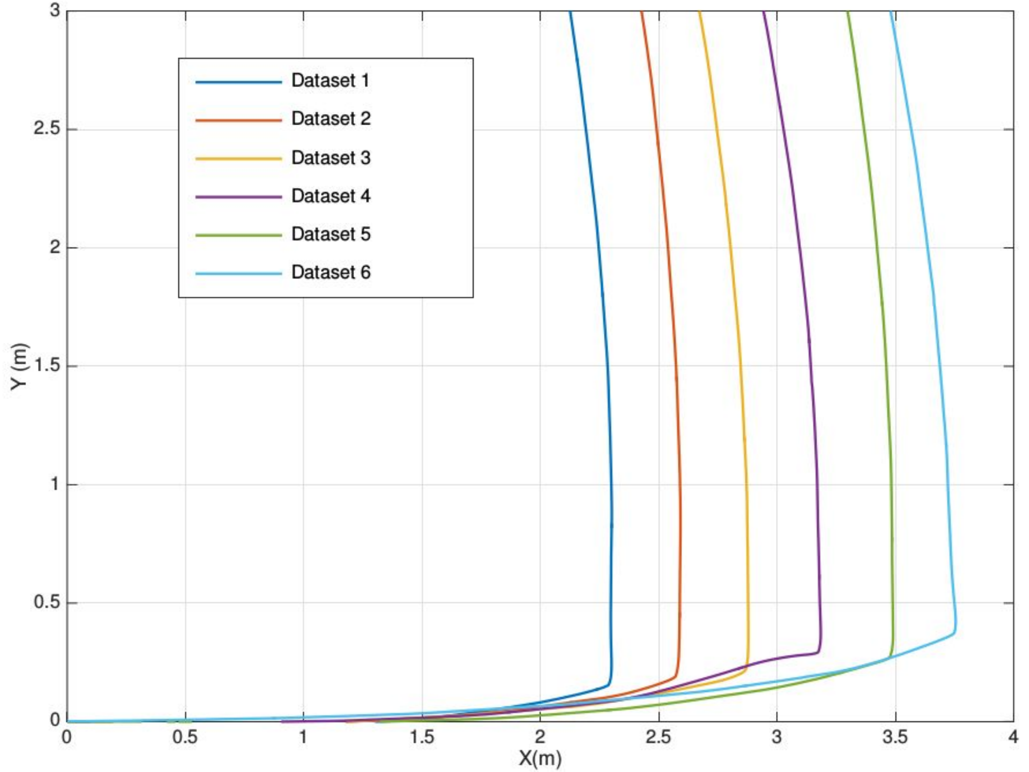
- We performed experiments inside a laboratory to establish the indoor localization capabilities of our system.
- The dimensions of the scene were **10m✕5m**

Indoor Experimentation

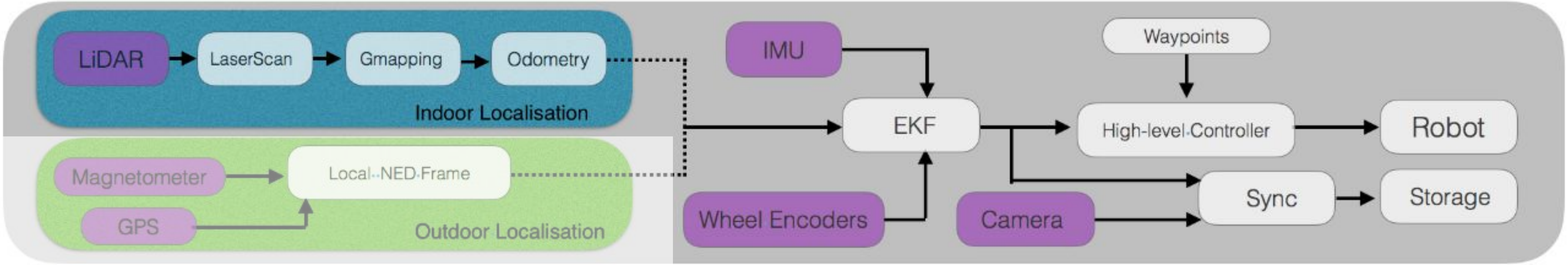
- We performed experiments inside a laboratory to establish the indoor localization capabilities of our system.
- The dimensions of the scene were **10m X 5m**
- Part of the region was well lit the other not so much, thus making a **challenging lighting** condition

Indoor Experimentation: Dataset Generation

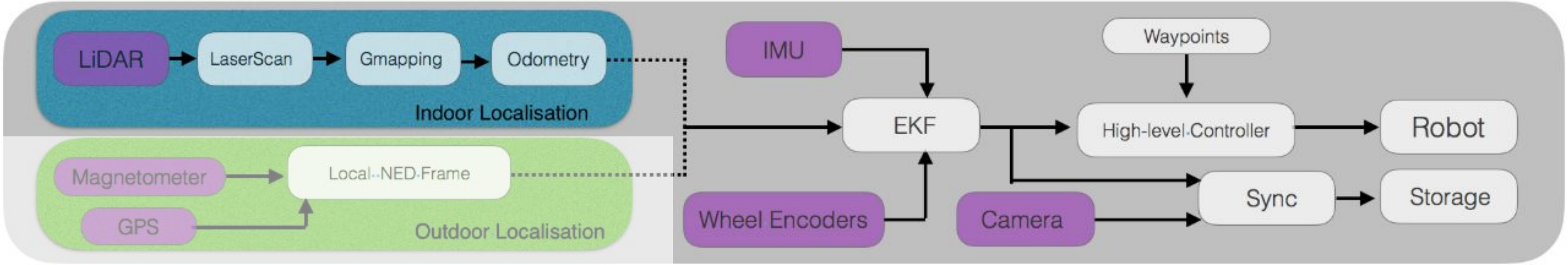
- Figure shows the paths used to generate the indoor dataset.
- **5717** images were collected in total for the paths shown.



Indoor Experimentation: Dataset Generation

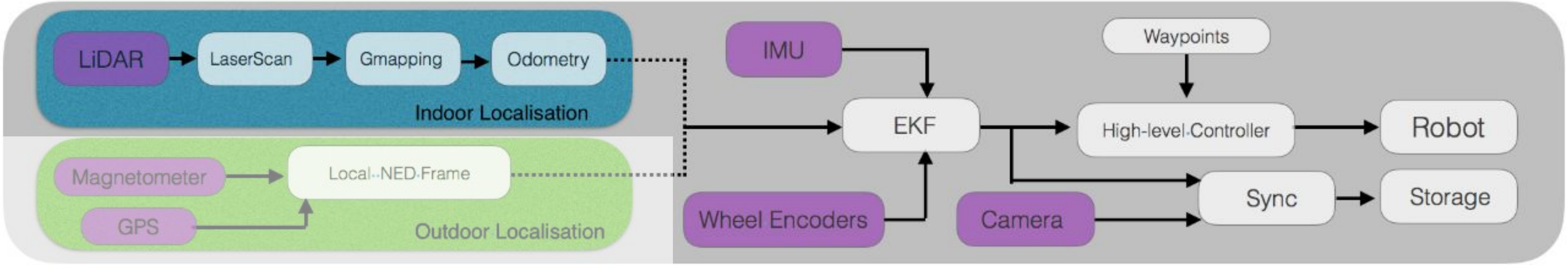


Indoor Experimentation: Dataset Generation



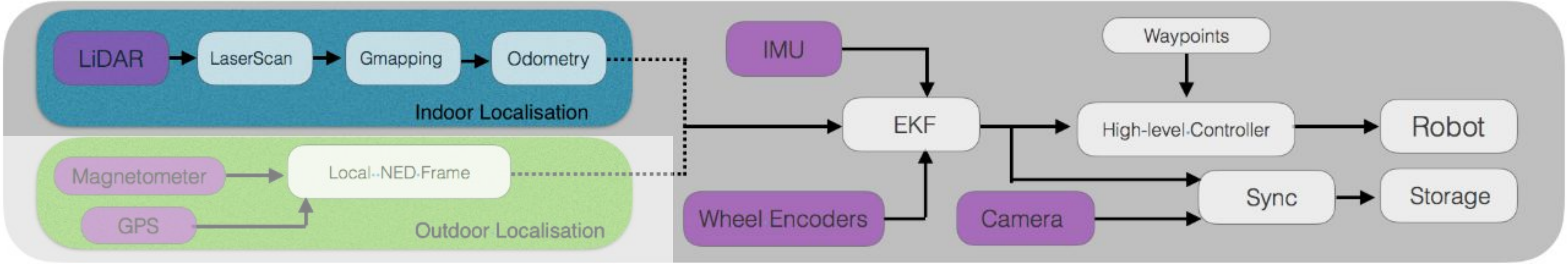
- The architecture used above is used for generating the ground truth.

Indoor Experimentation: Dataset Generation



- The architecture used above is used for generating the ground truth.
- We use open source implementation of `ros-gmapping` for performing **SLAM** on the Laser Scanner data and modified `robot-pose-ekf` for EKF.

Indoor Experimentation: Dataset Generation



- The architecture used above is used for generating the ground truth.
- We use open source implementation of `ros-gmapping` for performing **SLAM** on the Laser Scanner data and modified `robot-pose-ekf` for EKF.
- The **time synchronization** was done using an approximate time policy matching the time-stamps for different sensor readings.

Indoor Experimentation: Training

- We used **~4500** images for training in a **4:1** training and validation split and **~1000** for testing.
- We trained the network offline on an Nvidia GeForce **TITAN X** GPU.
- Trained for **50,000 iteration** in ~**6 hours**, avoiding overfitting with **validation** in every **100 iteration**.

Indoor Experimentation: Testing

- For testing we both **emulated** the performance and **deployed** on a robot.

Indoor Experimentation: Testing

- For testing we both **emulated** the performance and **deployed** on a robot.
- The network was deployed on an Nvidia Tegra TX1 for testing using open source software `ros-caffe`.
- The emulation on **TITAN X** runs at **200 Hz** whereas implementation on **TX1** runs at **18.5 Hz**.

Indoor Experimentation: Testing

- For testing we both **emulated** the performance and **deployed** on a robot.
- The network was deployed on an Nvidia Tegra TX1 for testing using open source software **`ros-caffe`**.
- The emulation on **TITAN X** runs at **200 Hz** whereas implementation on **TX1** runs at **18.5 Hz**.
- The average **error** we encountered were **0.38 ± 0.08 m** in the **10m×5m** environment.

Introduction

Motivation

Results

Methodology

Experiments

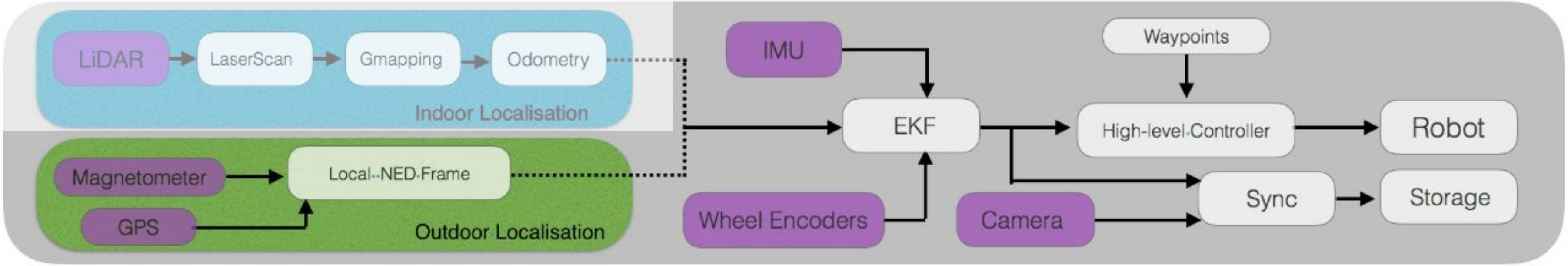
Indoor Experimentation

Outdoor Experimentation

Outdoor Experimentation

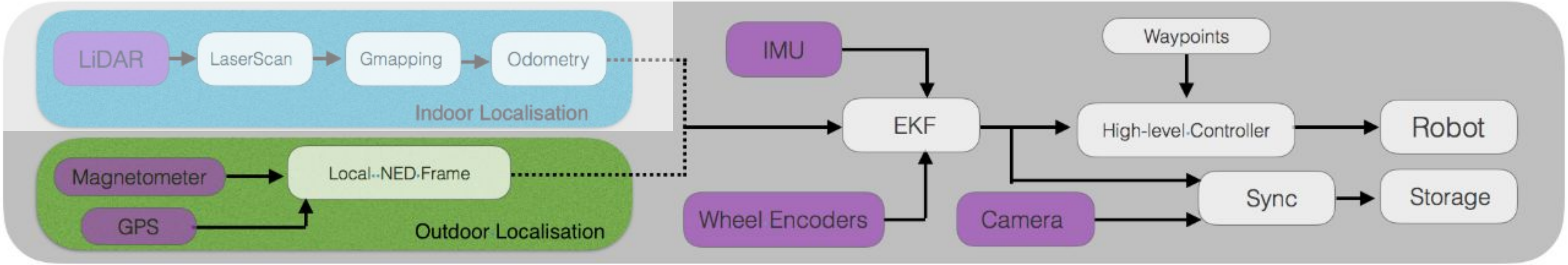
- Outdoor experiments were done on an empty road.
- The approximate dimensions of the scene were **50m⨉7m**, though we cover only **~30 m length** of this.
- We only created **straight line** datasets for this environment.

Outdoor Experimentation: Dataset Generation



- Similar to the indoor methodology.

Outdoor Experimentation: Dataset Generation



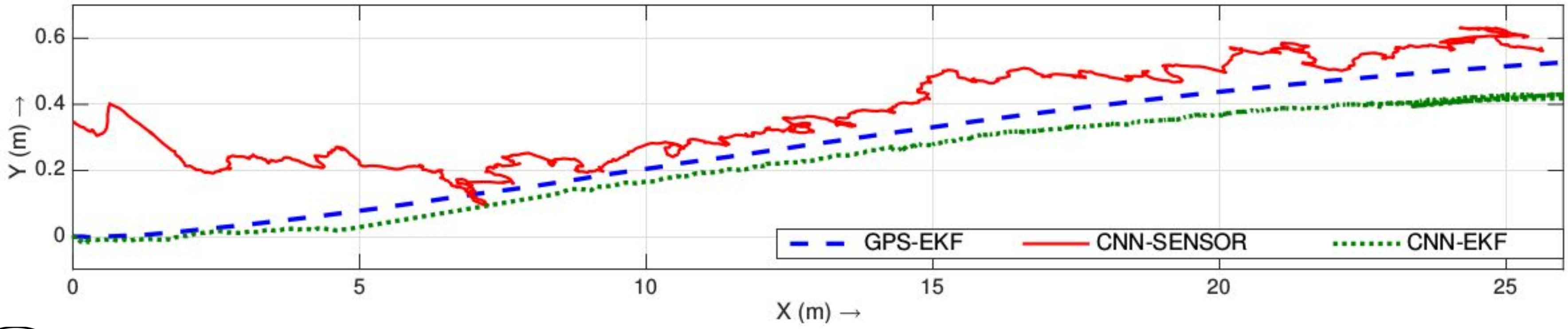
- Similar to the indoor methodology.
- Laser Scanner is replaced by **Magnetometer-GPS** sensor fusion for generating ground truth.

Outdoor Experimentation: Training

- We used **~6400** images recorded, out of which **~5000** images used in **4:1** split for training and validation. Rest of the images used for testing.
- Remaining details for training **same as for indoor one**.

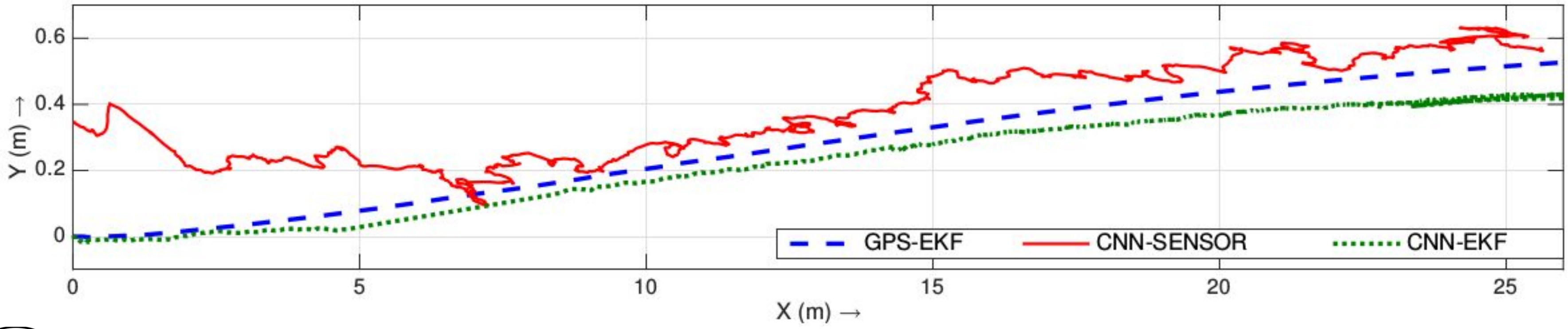
Outdoor Experimentation: Testing

- The testing method for outdoor experiment was same as the indoor one.
- Result from one of the runs is shown below.



Outdoor Experimentation: Testing

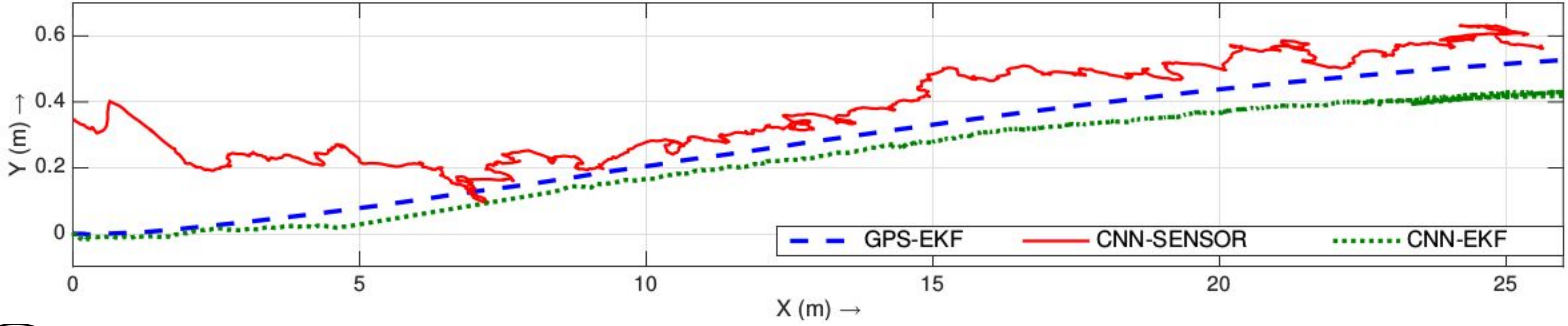
- The testing method for outdoor experiment was same as the indoor one.
- Result from one of the runs is shown below.
- The outdoor relocalization **error** we encountered without fusion were **2.01 ±0.90 m** in the **50m✕7m** environment.



IIT
Kanpur

Outdoor Experimentation: Testing

- The testing method for outdoor experiment was same as the indoor one.
- Result from one of the runs is shown below, **notice error in raw estimate.**
- The outdoor relocalization **error** we encountered without fusion were **2.01 ±0.90 m** in the **50m✕7m** environment.



Introduction

Motivation

Results

Methodology

Experiments

Indoor Experimentation

Outdoor Experimentation

Motivation

Results

Methodology

Experiments

Indoor Experimentation

Outdoor Experimentation

Failure Condition

Failure Conditions

- Since we were using **shallower networks** than methods which perform similar task and **cheaper sensors**, some kinds errors were **expected**.

Failure Conditions

- Since we were using **shallower networks** than methods which perform similar task and **cheaper sensors**, some kinds errors were **expected.**

1. **Blurring** of images at high speed.

Failure Conditions

- Since we were using **shallower networks** than methods which perform similar task and **cheaper sensors**, some kinds errors were **expected**.

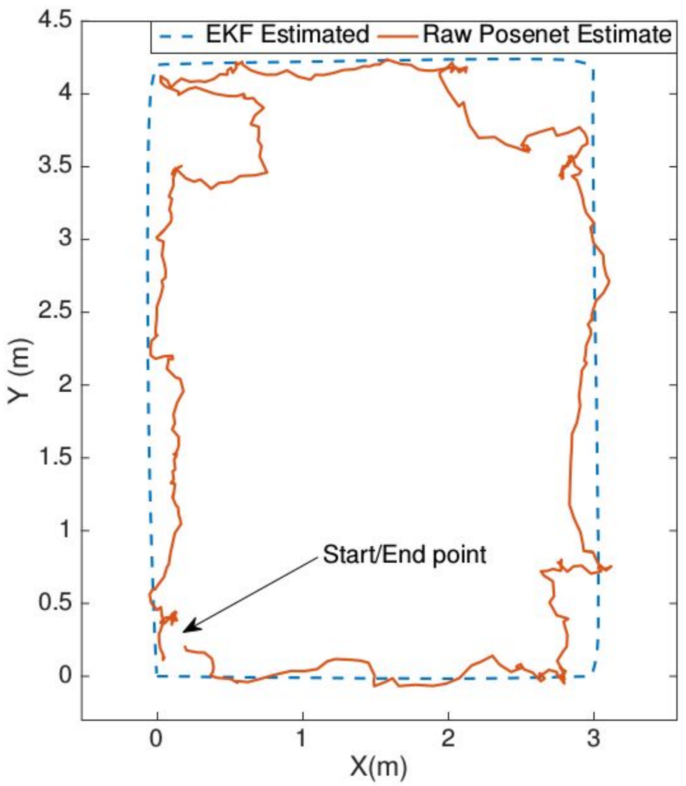
1. **Blurring** of images at high speed.
2. Larger relocalization error in **repetitive** environments.

Failure Conditions

Blurring of images at high speed.

This affected us only at the **turns** at the changes are high then.

Turns were especially bad as we were **not regressing angle** values from the network only position.



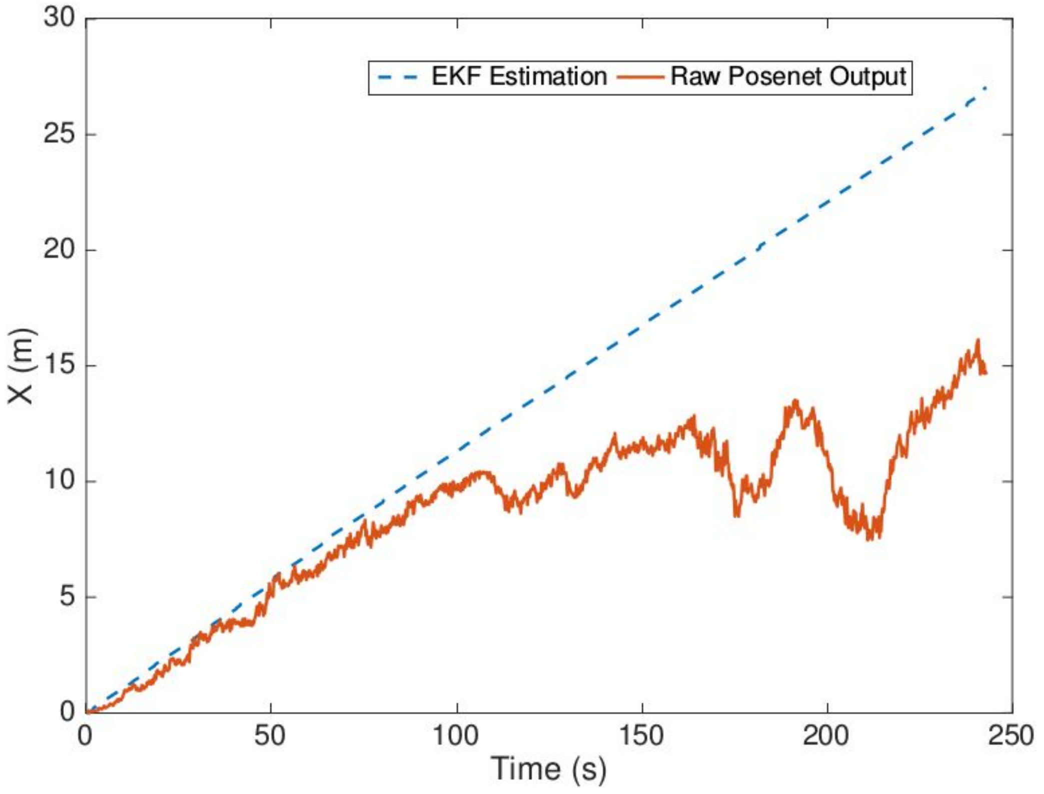
IIT
Kanpur

Failure Conditions

Larger relocalization error in **repetitive** environments.

This error occurs due to network's **inability to differentiate** between different **locations**.

Possible fix might be to add more layers.



IIT Kanpur

Introduction

Motivation

Results

Methodology

Experiments

Implementation Details

IIT
Kanpur

Introduction

Motivation

Results

Methodology

Experiments

Implementation Details

IIT Kanpur

Motivation

Methodology

Results

Experiments

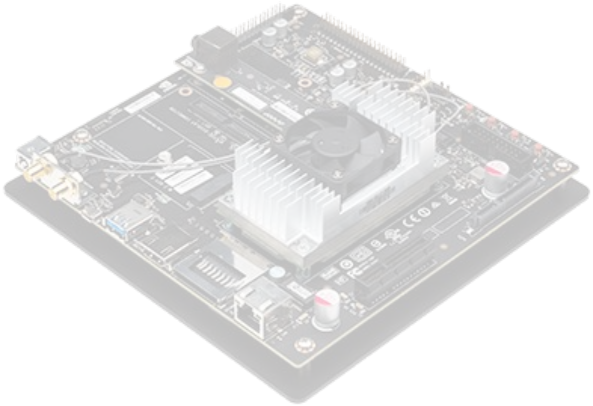
Implementation Details

Hardware

Software

IIT
Kanpur

Indoor



Onboard
Computer ?

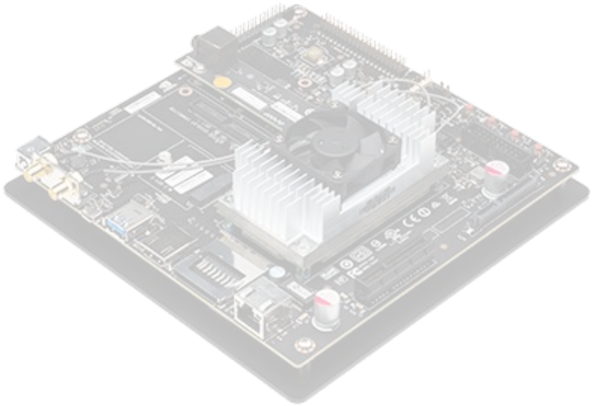
Robotic Platform ?

IMU ?

Camera ?

IIT
Kanpur

Indoor



Onboard
Computer ?

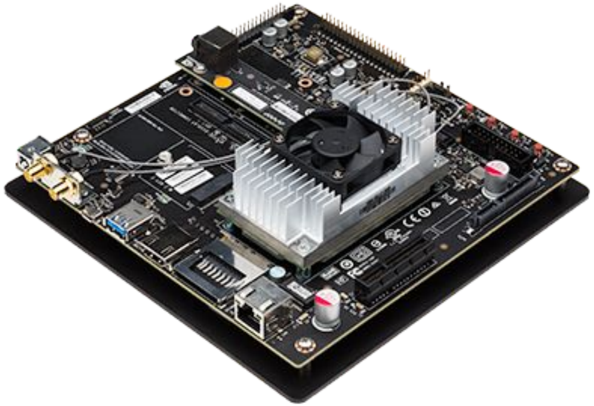
Nex Robotics FireBird IV
+
Wheel Encoders

IMU ?

Camera ?

IIT
Kanpur

Powered with one 3 cell, 2400mAH LiPo Battery, runtime >10 min.



Indoor

Nvidia Tegra TX1
Development Board

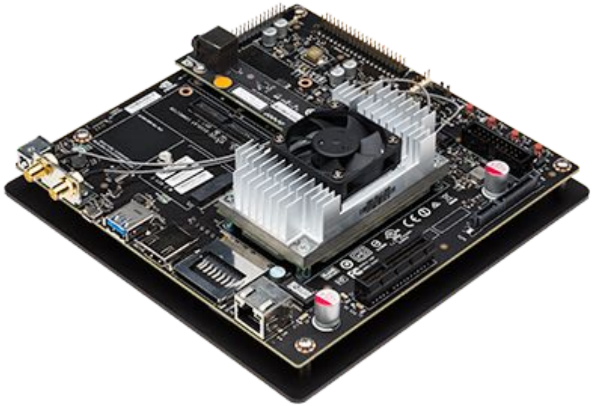
Nex Robotics FireBird IV
+
Wheel Encoders

IMU ?

Camera ?

ARM Processor, 256 CUDA Cores, 4 GB RAM

Indoor



Nvidia Tegra TX1
Development Board

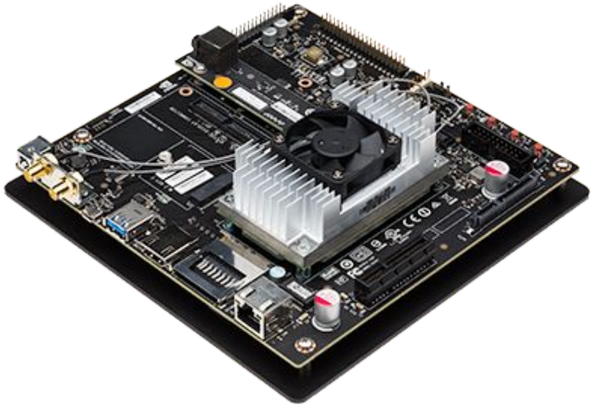
PixHawk

Nex Robotics FireBird IV
+
Wheel Encoders

Camera ?

IIT
Kanpur

Indoor



Nvidia Tegra TX1
Development Board

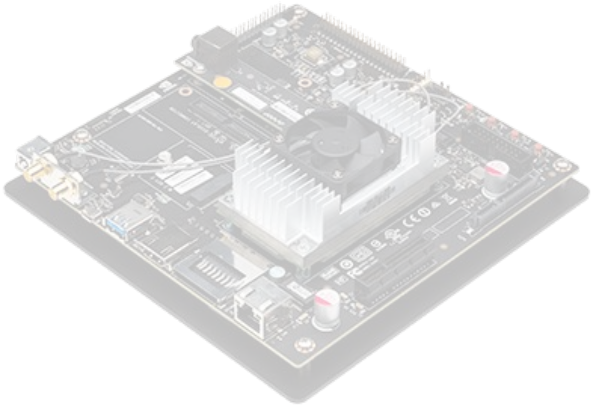
PixHawk

Nex Robotics FireBird IV
+
Wheel Encoders

Logitech C270

IIT
Kanpur

Outdoor



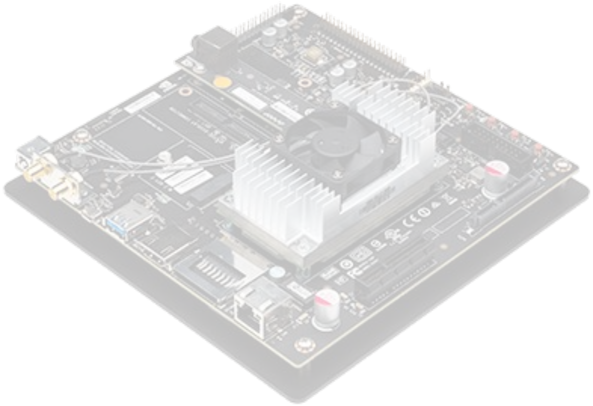
Onboard Computers ?

Robotic Platform ?

IMU ?

Camera ?

Outdoor



Onboard
Computers ?

Nex Robotics 0xDelta
+
Wheel Encoders

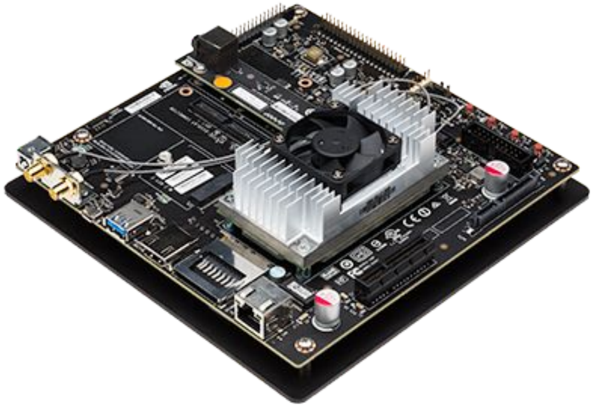
IMU ?

Camera ?

IIT
Kanpur

Powered with one 6 cell, 10,000mAH LiPo Battery, runtime >30 min.

Outdoor



Nvidia Tegra TX1
Development Board
+
Intel NUC

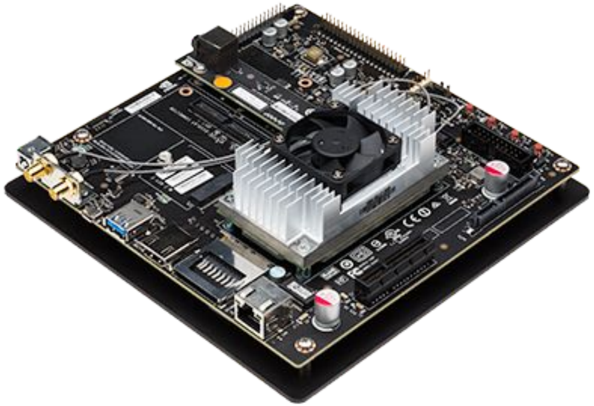
Nex Robotics 0xDelta
+
Wheel Encoders

NUC: i5, 8 GB RAM

IMU ?

Camera ?

Outdoor



Nvidia Tegra TX1
Development Board
+
Intel NUC

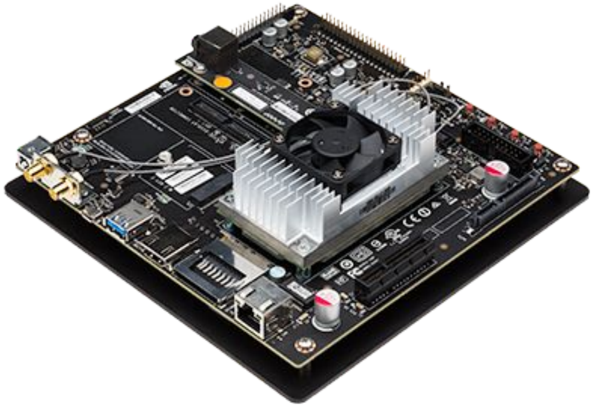
Nex Robotics 0xDelta
+
Wheel Encoders

PixHawk IMU + GPS

Camera ?

IIT
Kanpur

Outdoor



Nvidia Tegra TX1
Development Board
+
Intel NUC

Nex Robotics 0xDelta
+
Wheel Encoders

PixHawk IMU + GPS

Genius Widecam

IIT
Kanpur

Motivation

Methodology

Results

Experiments

Implementation Details

Hardware

Software

IIT
Kanpur

Implementation Details

Operating System

Platform

Platform/Library

Library

ubuntu

Caffe2

OpenCV

Implementation Details

ubuntu

Platform

Platform/Library

Library

Implementation Details

Platform/Library

Library

Implementation Details

Library

Implementation Details

ubuntu

KINETIC KAME ROS

Caffe2

OpenCV

Implementation Details

More Open Source Libraries/Implementations Used:

- `ros-gmapping`
- `mavros`
- `ros-caffe`
- `robot-pose-ekf`

CNN Sensors: Implementation Details

Other details of the network

Loss Function : $\mathcal{L}_{\mathbf{x}} = \left\|\hat{\mathbf{x}} - \mathbf{x}\right\|_2$, where **x** is the regression value and $\hat{\mathbf{x}}$ is the ground truth.

Optimizer: Adam and Adagrad

Initialization: Xavier

What’s Next ? Or
What else can one do with this ?

Future!

- Introduce the **regression** of **full 6d Pose**, **position** and **quaternion**.
- Add image processing based methods for **motion blur removal** to the pipeline.
- If the two above are done, compress the network and use the CNN-EKF on a **Quadrotor**!
- Improve the performance on moderately repetitive environments by modifying the architecture of CNN Sensor.

Questions ??

Acknowledgements

Thanks!

- This project was in part funded by the Center for Artificial Intelligence & Robotics (CAIR), DRDO, Bangalore, India


THANK
YOU
GRACIAS
ARIGATO
SHUKURIA
DANKSCHEEN
BIYAN
SHUKRIA
TINGKI
YAQHANYELAY
TASHAKKUR ATU
SUKSAMA
EKHMET
MEHRBANI
GRAZIE
MAAKE
KOMAPSUMNIDA
PALDIES
BOLZÏN
MERCI
GOZAIMASHITA
EFCHARISTO
JUSPAXAR
SPASSIBO
SNACHALHUYA
NUHUN
CHALTU
WABEEJA
MAITEKA
SPASIBO
DENKAUJA
NENACHALHYA
MERASTAWHY
GAEJTHO
TAVTAPUCH
MEDAWAGSE
AGUYJE
FAKAAUE
SIKOMO
MINMONCHAR
ATTO
LAH