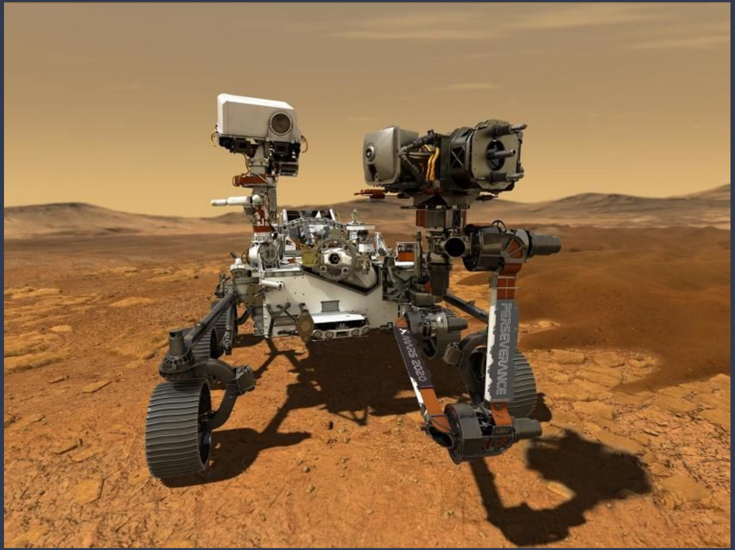
Real Time Visual Localization And Mapping

Nischal Maharjan	073 BEX 421
Rashik Shrestha	073 BEX 432
Sajil Awale	073 BEX 436
Shrey Niraula	073 BEX 443



Perseverance Rover by NASA

Landed on Mars on Feb. 18, 2021

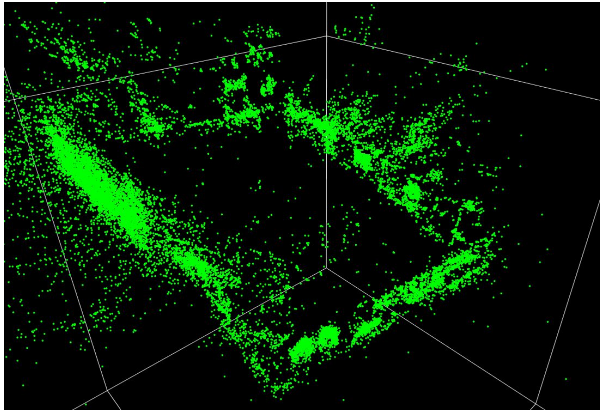
But how it will navigate on totally unknown environment ?

Image Source:
https://www.pcmag.com/news/nasas-mars-perseverance-rover-landing-how-to-watch-and-whats-on-board

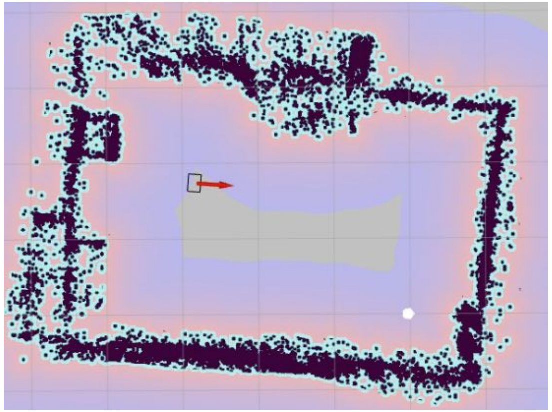
What?

Is the project about

What?



Map



Localize

Deal with moving people

Using Visual Sensors Only

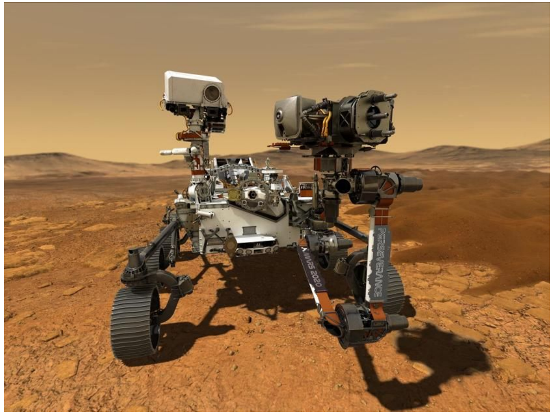
Why?

The project has been done

Why?

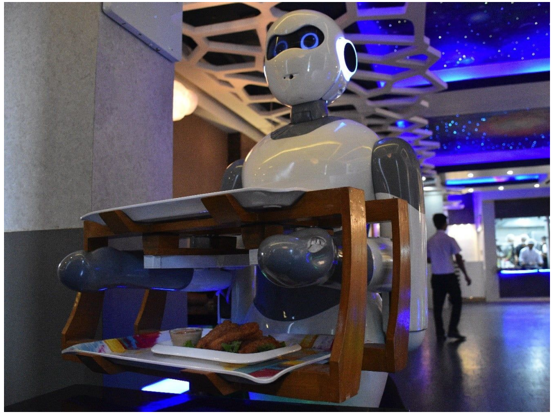
Self Driving Cars

Image Source
https://en.wikipedia.org/wiki/Self-driving_car



Unmanned Vehicles

Image Source:
https://www.pcmag.com/news/nasas-mars-perseverance-rover-landing-how-to
-watch-and-whats-on-board



Autonomous Navigation

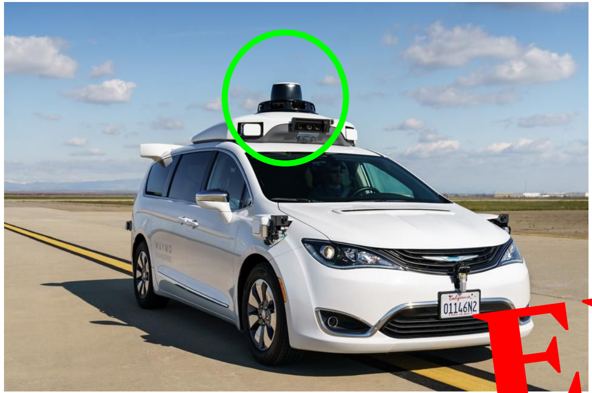
Image Source:
https://www.digitaltrends.com/cool-tech/robot-waiter-ginger/

How?

The project was done

How?

Popular Visual Sensors



LIDAR

Depth Camera

Stereo Camera

TOO EXPENSIVE

Image Source: https://www.forbes.com/sites/alanohnsman/2019/04/23/teslas-elon-musk-trashes-lidar-for-self-driving-cars-but-waymo-is-rolling-out-a-new-one/?sh=2259e8c85a9d

Image Source: https://jahya.net/blog/how-depth-sensor-works-in-5-minutes/

Image Source: https://www.amazon.ca/MYNT-Stereo-Camera-Depth-Sensor/dp/B07NJ4GL6X

Why?

"Lidar is a fool's errand, anyone relying on lidar is doomed. Doomed! "

- Elon Musk

CEO, and product architect of Tesla

How?

Monocular cameras are the cheap option

But, it needs more computational power to achieve same accuracy as expensive sensors

How?

Our Approach

Single Monocular Camera

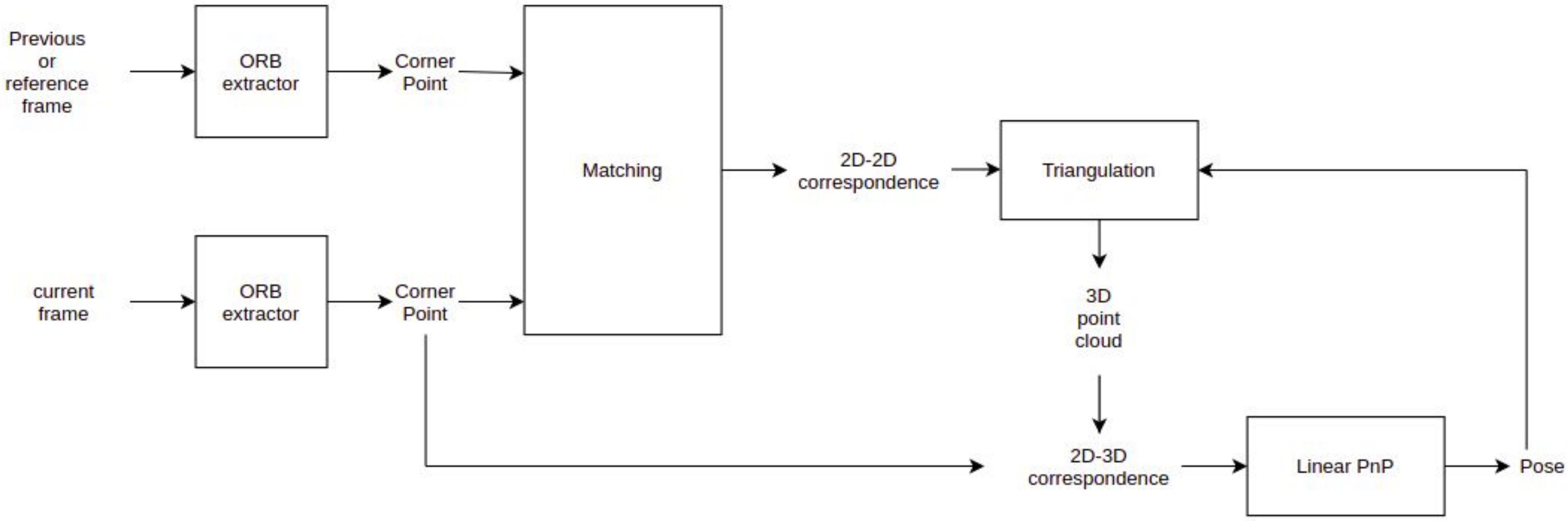
➕

Limited Computational Power

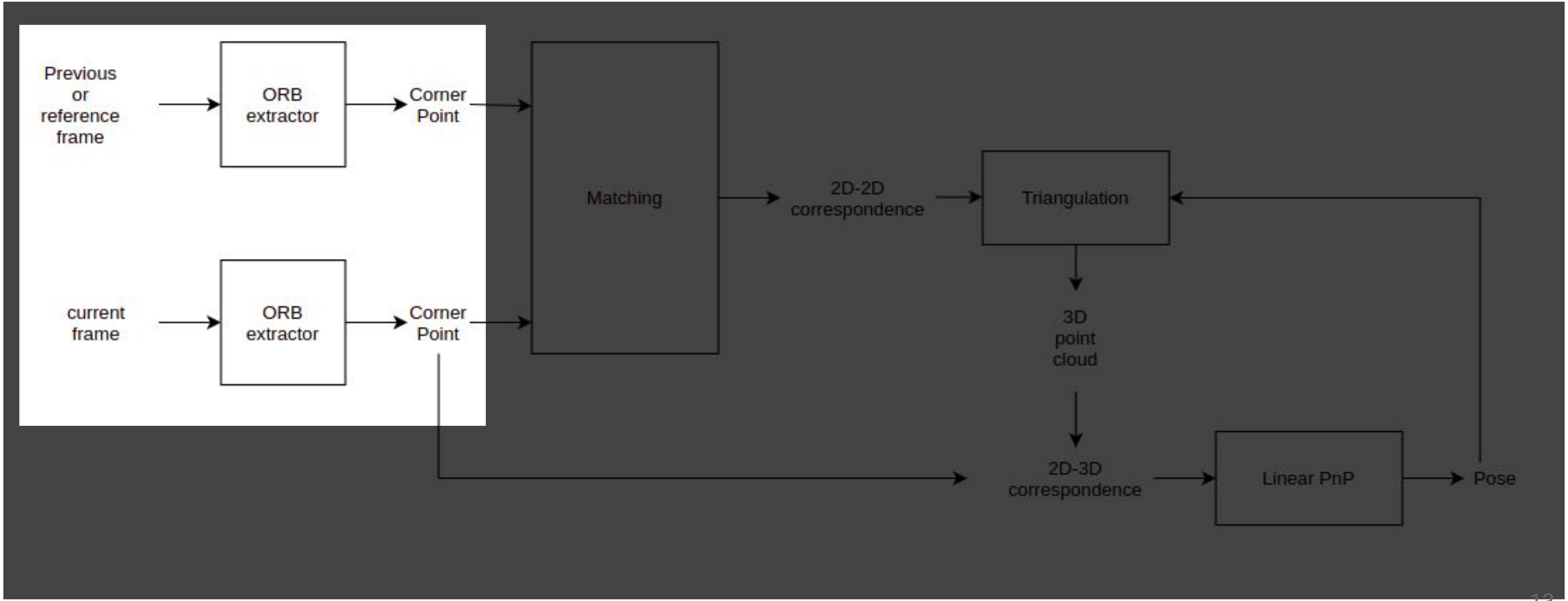
(CPU only Computation)
(No GPU acceleration)

Using Visual SLAM

Structure from Motion Paradigm

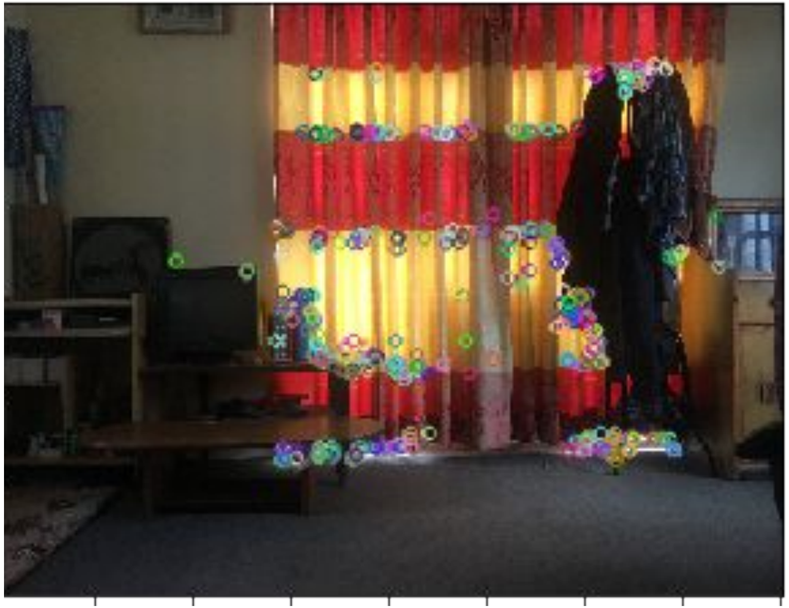


ORB Extraction

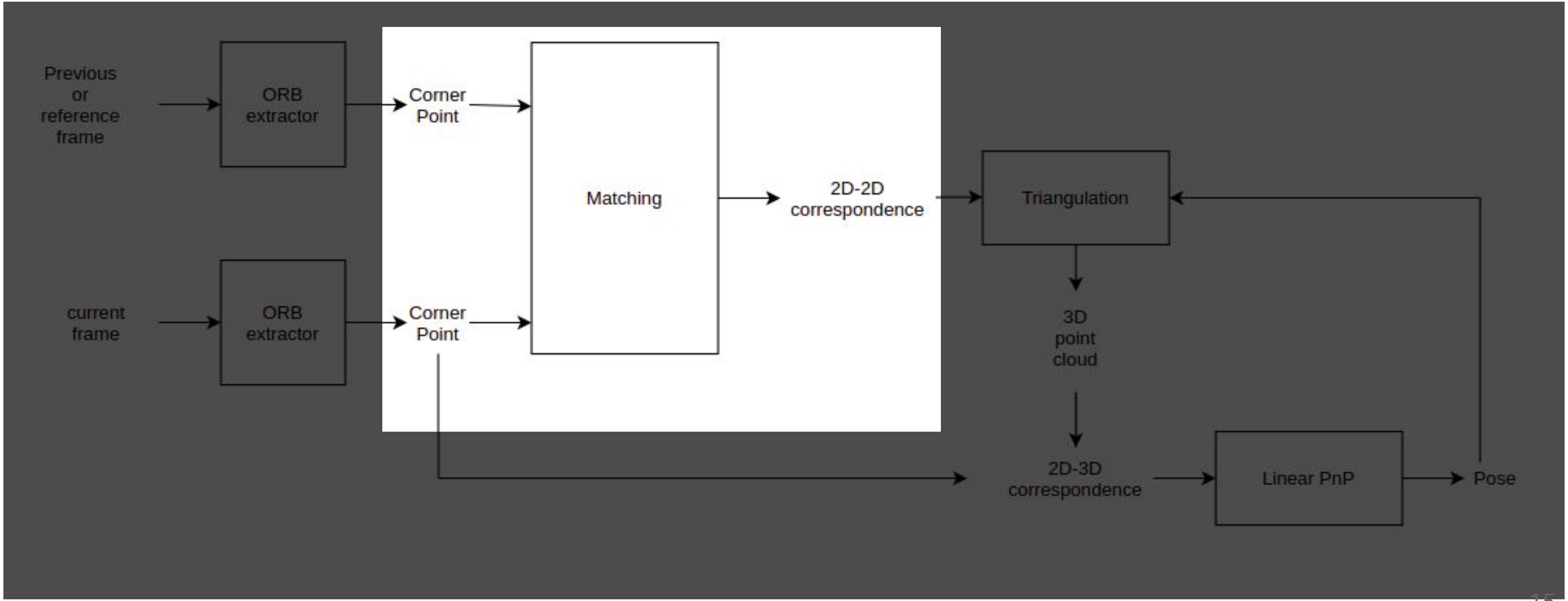


Original Image

Corner points detected



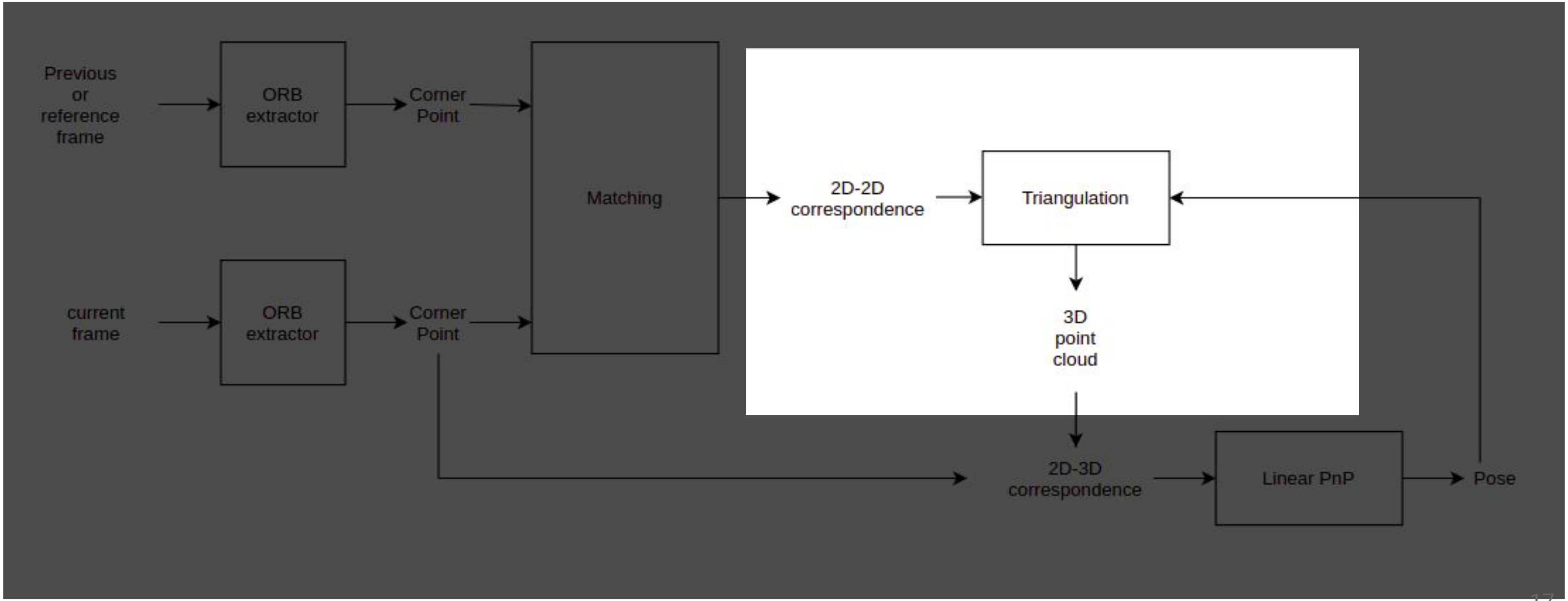
Feature Matching



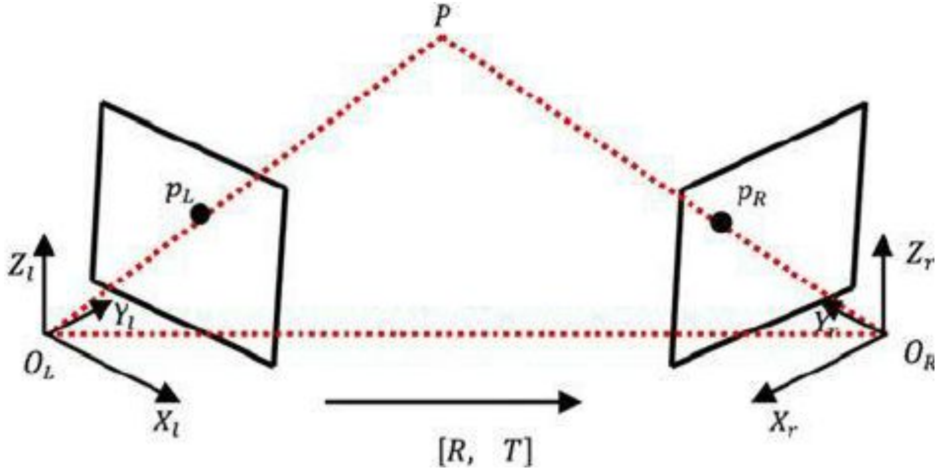
Keypoint Matches with number of outliers

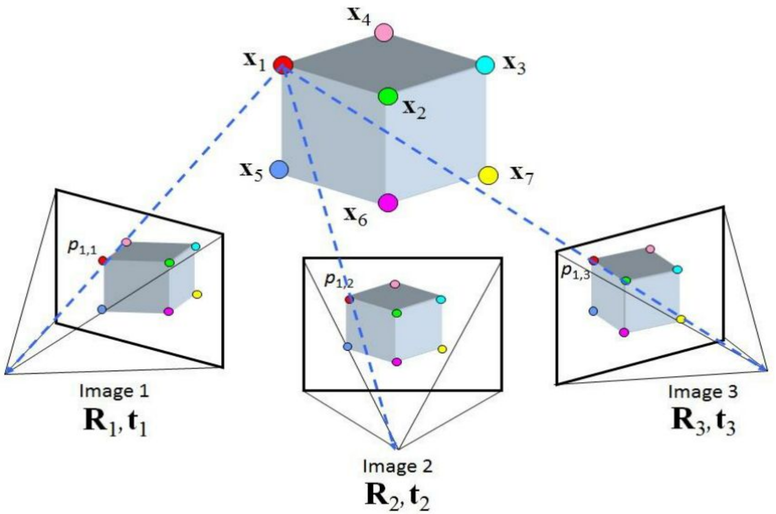
Keypoint matches after selecting inliers satisfying epipolar constraint using RANSAC

Triangulation

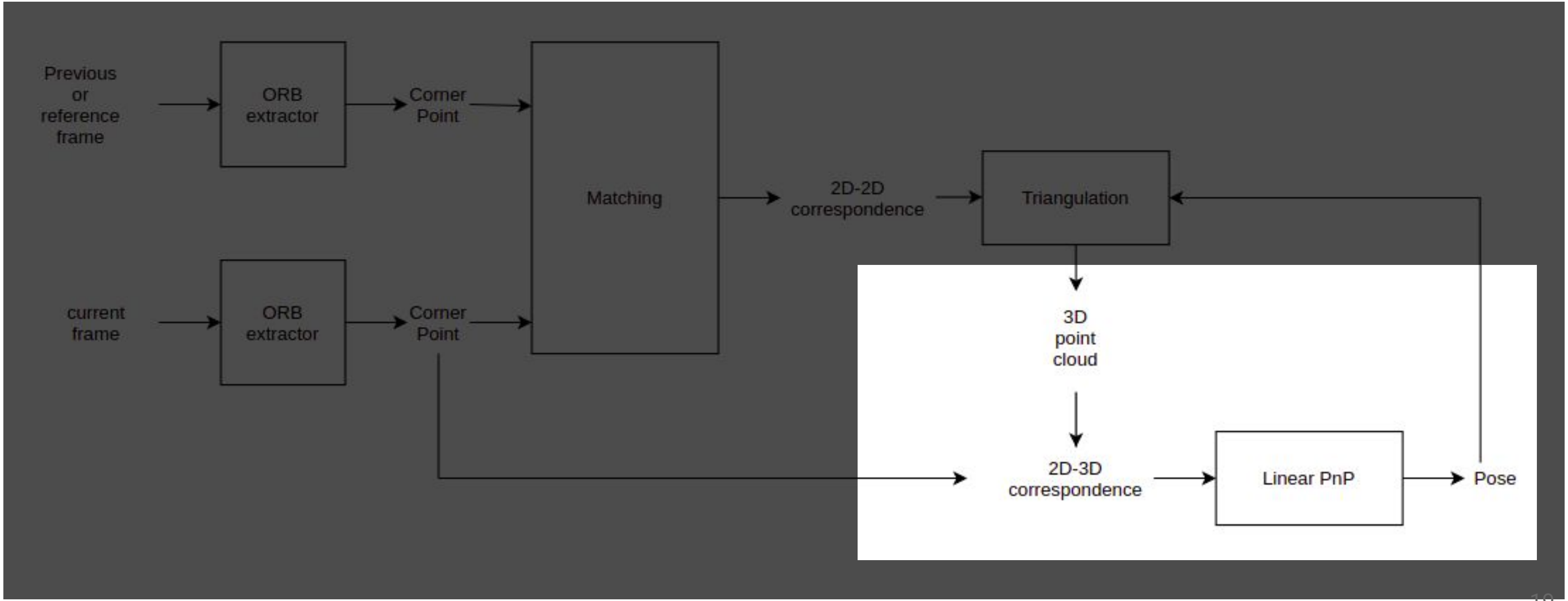


Given 2D-2D correspondence and relative pose between two images respective 3D point is estimated

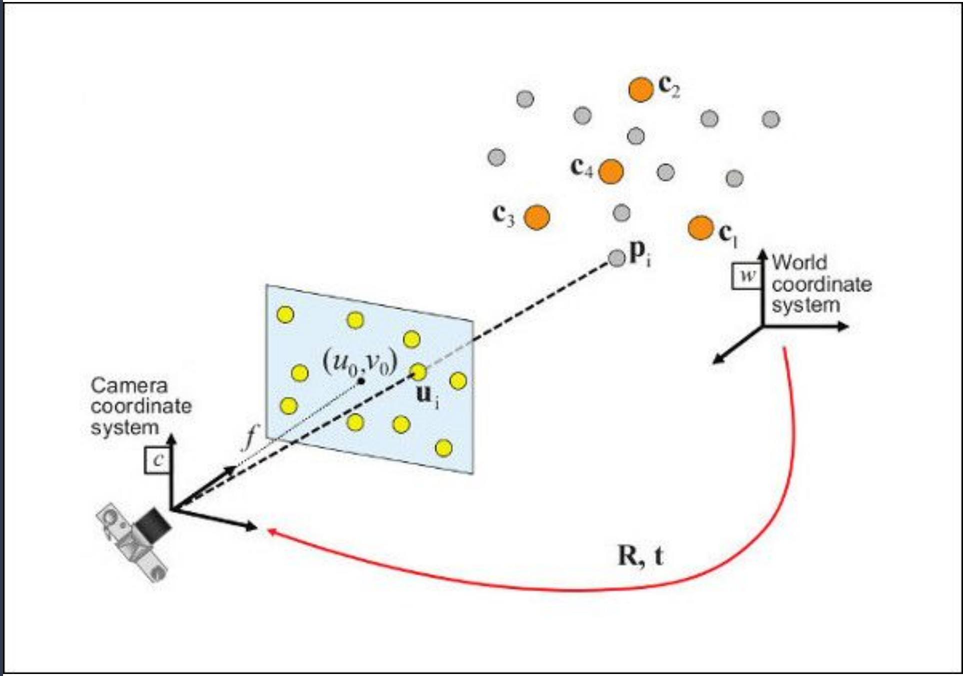




Linear PnP(Pose Estimation)



Given 2D-3D correspondence between image and 3d point cloud relative pose of image wrt world coordinate system can be estimated



Mapping :

Triangulation Generates 3D point cloud The generate local point cloud are stitched together to generate the map.

Localization:

Linear Pnp estimates the pose of camera in the 3D world coordinate system.

The pose generated by Linear PnP is used as input for the triangulation and the 3D point cloud generated is used to determine 3D-2D correspondence for pose estimation using Linear PnP. These two process of Map generation and pose estimation occurs in hand in hand simultaneously. Thus termed as SLAM(Simultaneous Localisation and Mapping)

Graph Optimization

Graph Optimization

Measurements collected

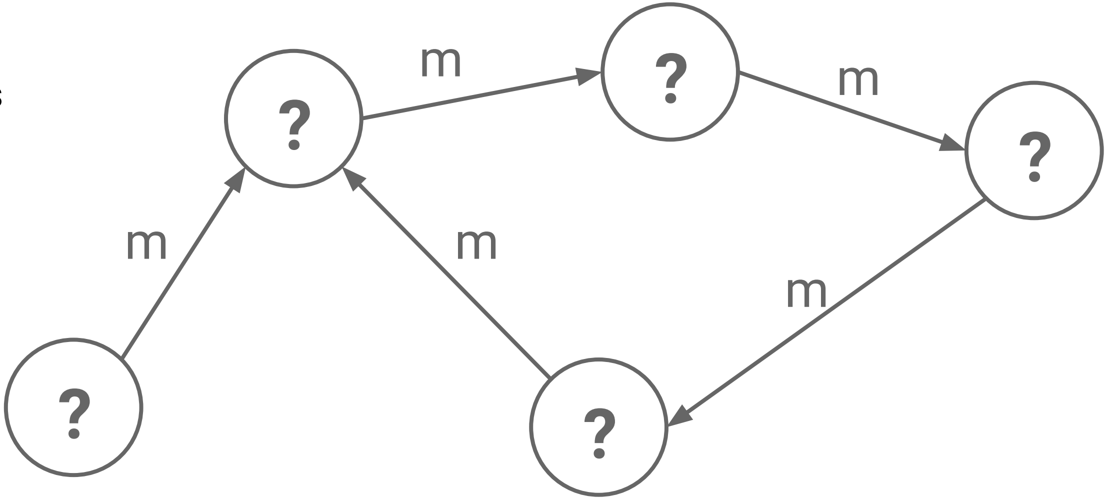
1. Relative transformation between adjacent robot poses
2. 3d coordinates of points in point cloud

But measurements are affected by Noise

Graph Optimization

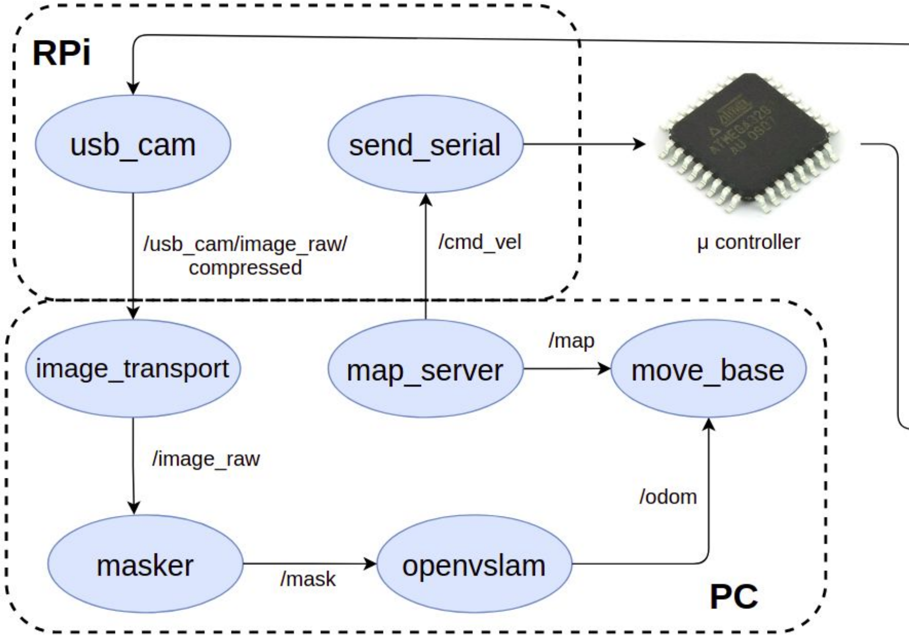
Nodes represents Robot States

Edges represents measurements

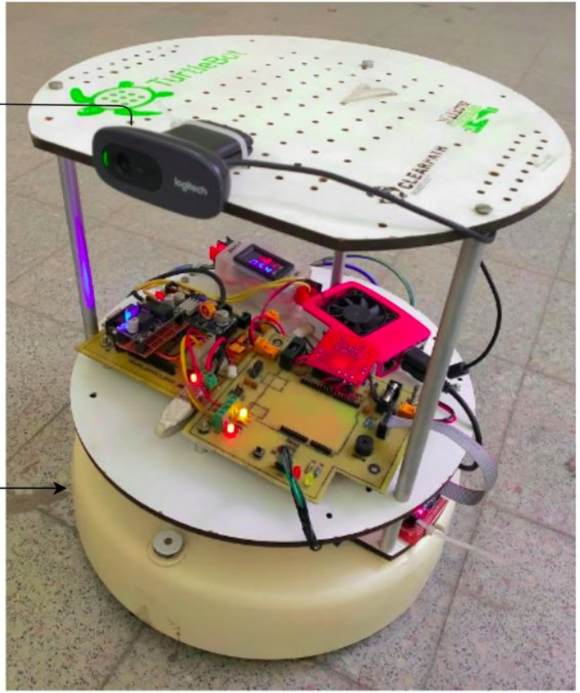


Goal:
Find the set of Robot states that maximizes the likelihood of given measurements affected by Gaussian noise

Communication Architecture



ROS Architecture

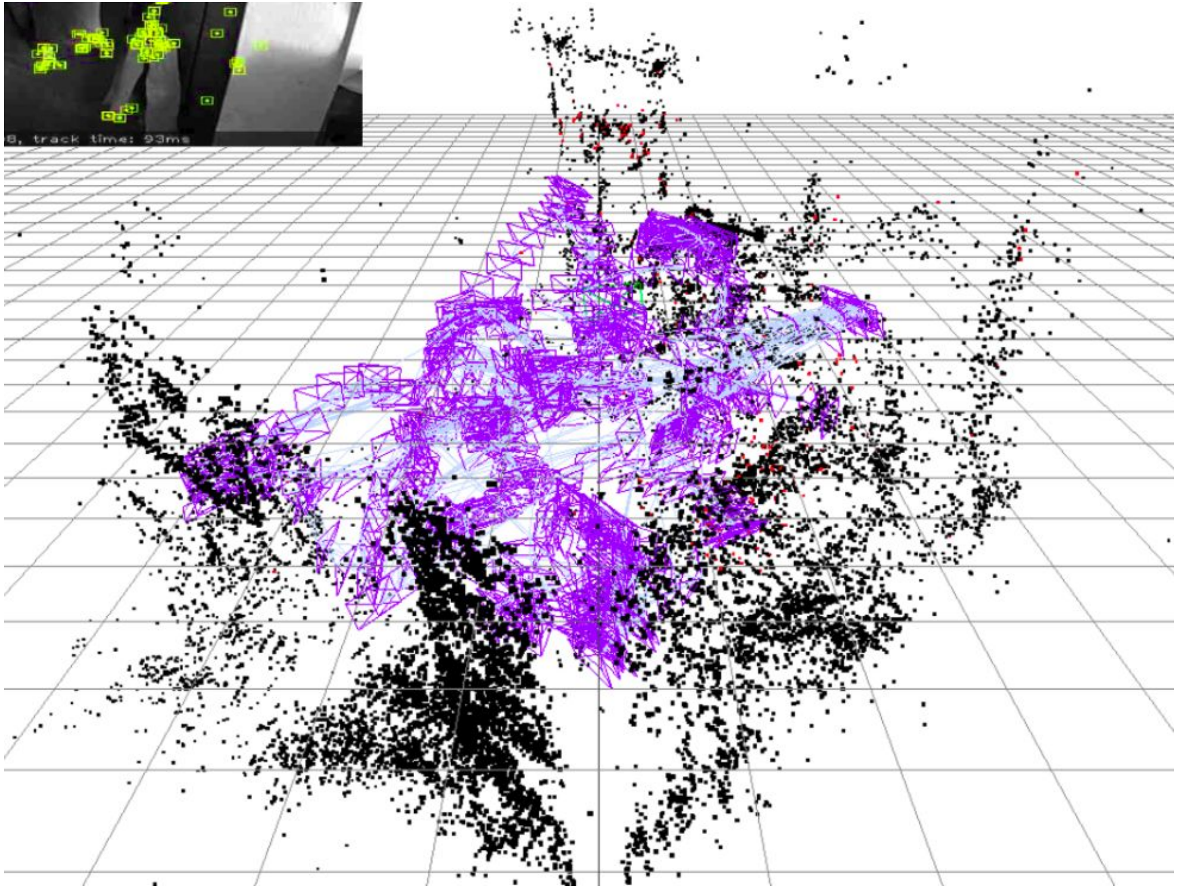


Mobile Robot

Mapping

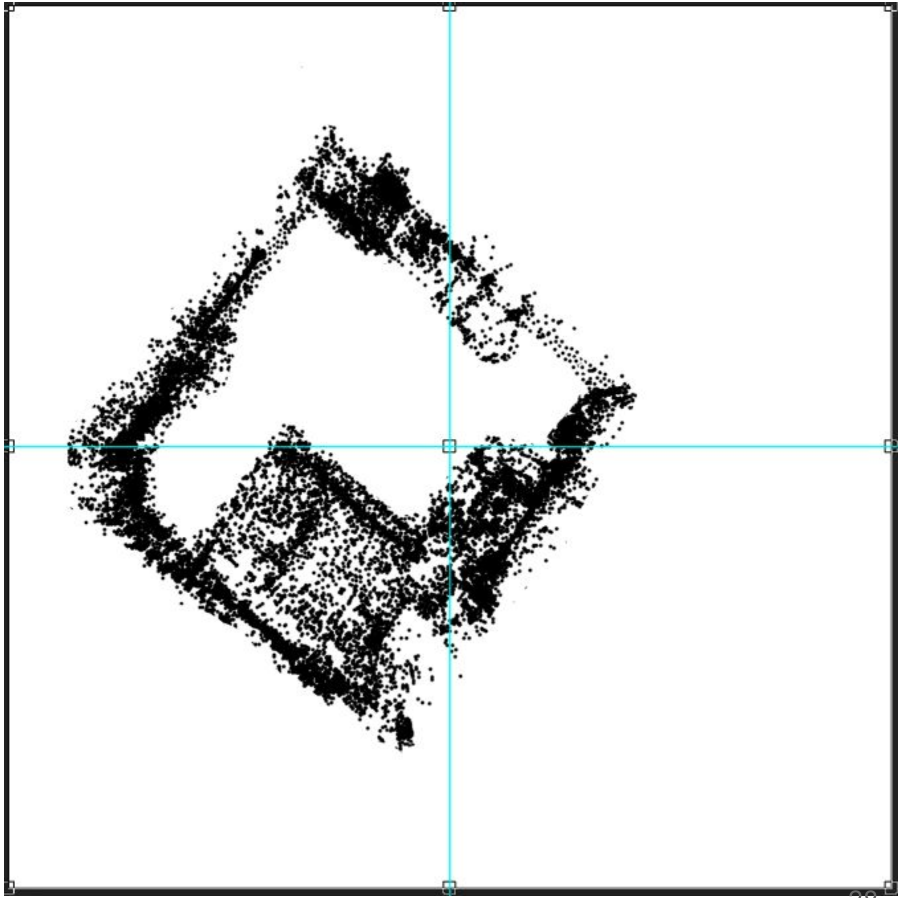
Storing the information about surrounding in memory

3D map of a room



Occupancy grid map

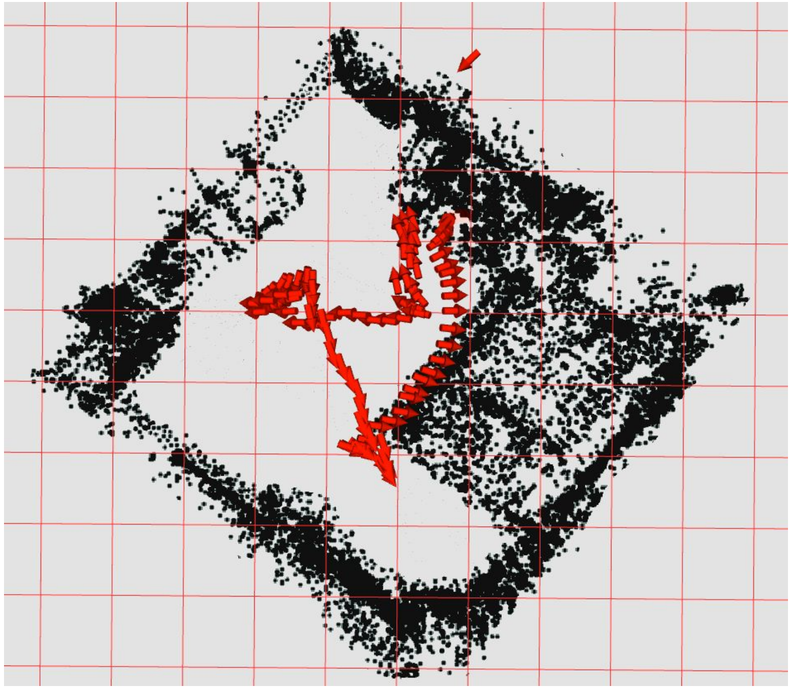
- 2D projection of 3D map
- Unwanted points are manually filtered



Localization

Finding your pose with respect to the prebuilt map

Visualizing live odometry

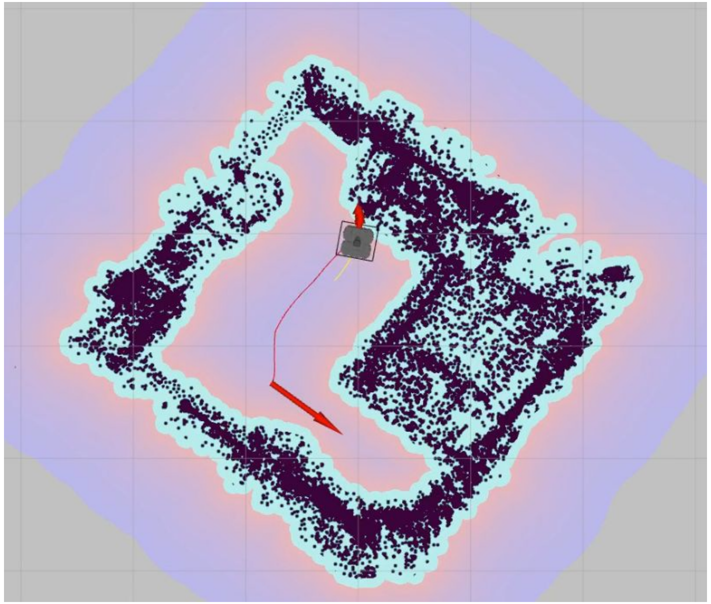


Navigation

Planning path from current position to destination

Path planning

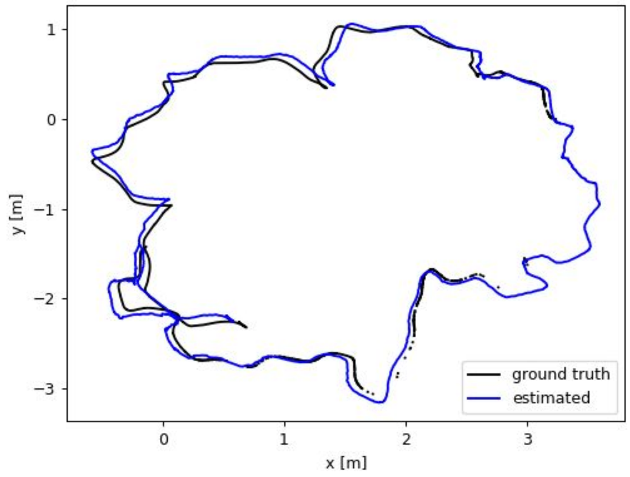
- Used to find best route from current location to destination
- Uses A* algorithm

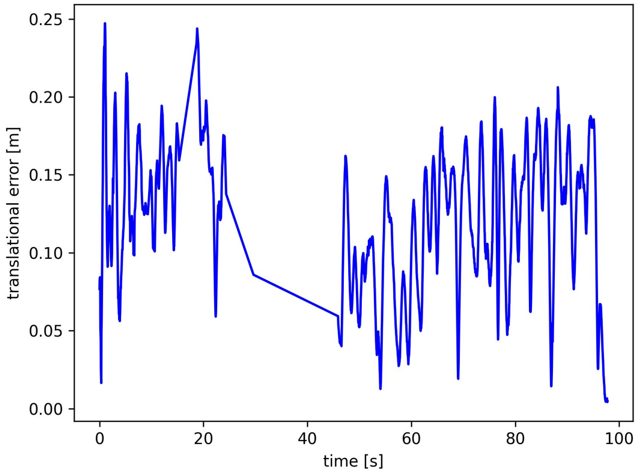


Static Environment Datasets

fr2_ desk dataset

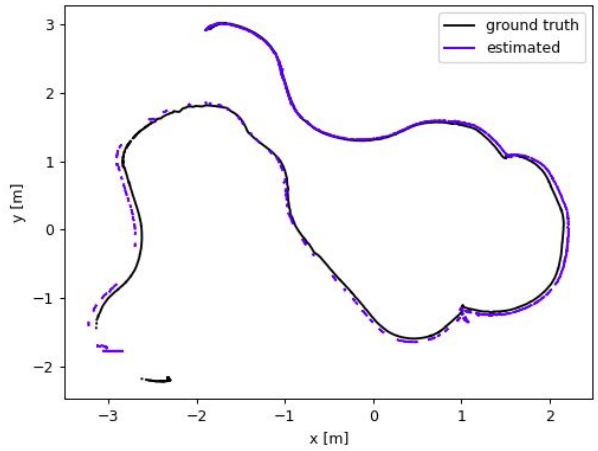
RMS error: 9.7710 cm
Relative Translational error: 12.9474 cm
Relative Rotational error: 14.37 degree

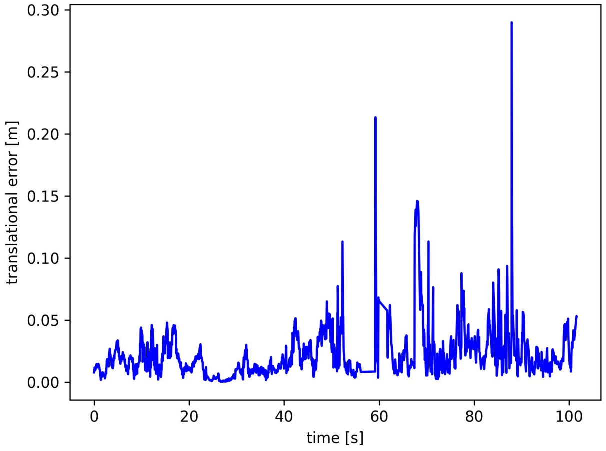




fr2_pioneer_slam2 dataset

RMS error: 10.196 cm
Relative Translational error: 2.8162 cm
Relative Rotational error: 1.059033 degree





Localization issues

Problems due to **dynamic objects** in the environment

Dynamic Obstacle Avoidance

Dynamic Obstacle Avoidance

- Dynamic Objects: Human, Vehicles, Animals
- Causes problem while mapping and tracking

- Map corrupted due to their inclusions

- Key Points from them to be removed

How to tackle dynamic object then... ?

Segmentation is chosen as method due to:

- Easy availability of pretrained models
- Availability of dataset with labels

Among segmentation methods, we prefer to go for **Semantic Segmentation** method because:

- Faster segmentation method
- Has **High speed** models for even **CPU** (ICNet)

How Does masking Help??

- Reduction of **tracking error**

- Removal of **Keypoints**

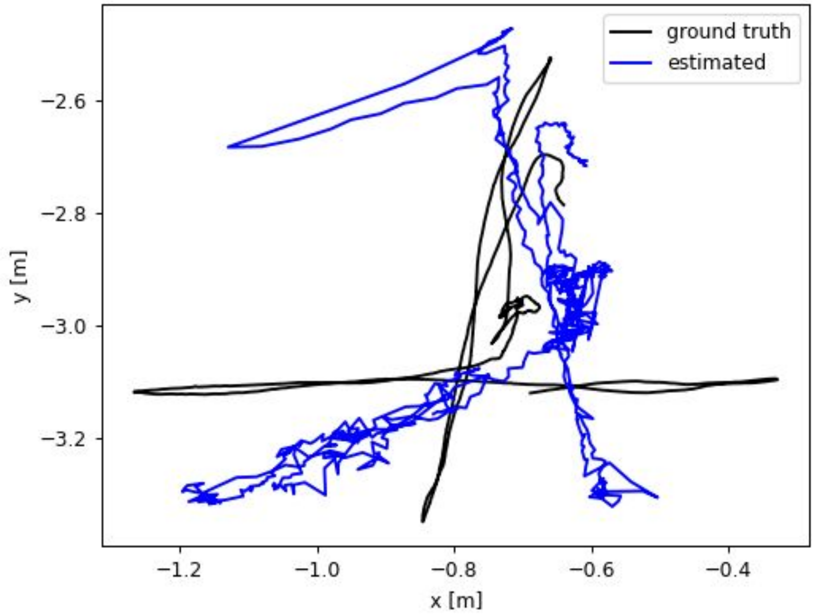
Removal of Keypoints from Dynamic Object

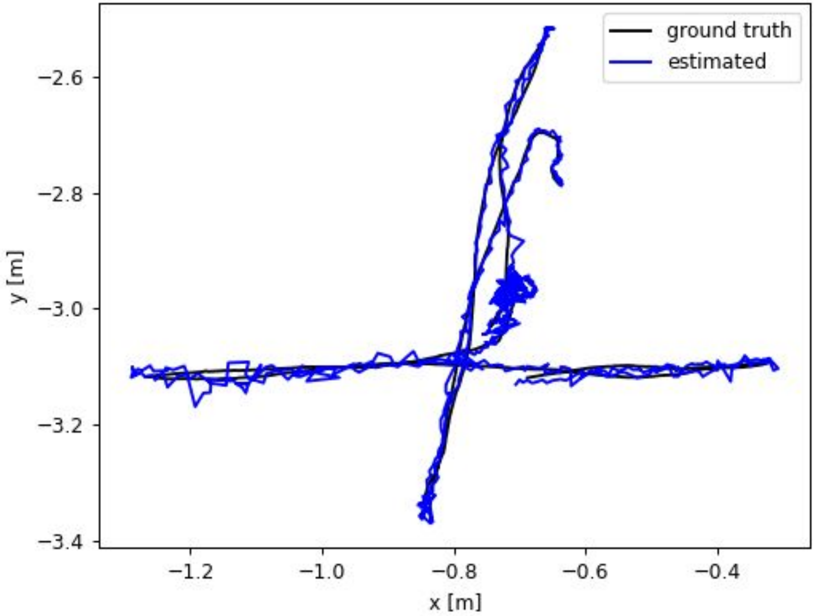
Figure 4.10: Before masking

Figure 4.11: Mask

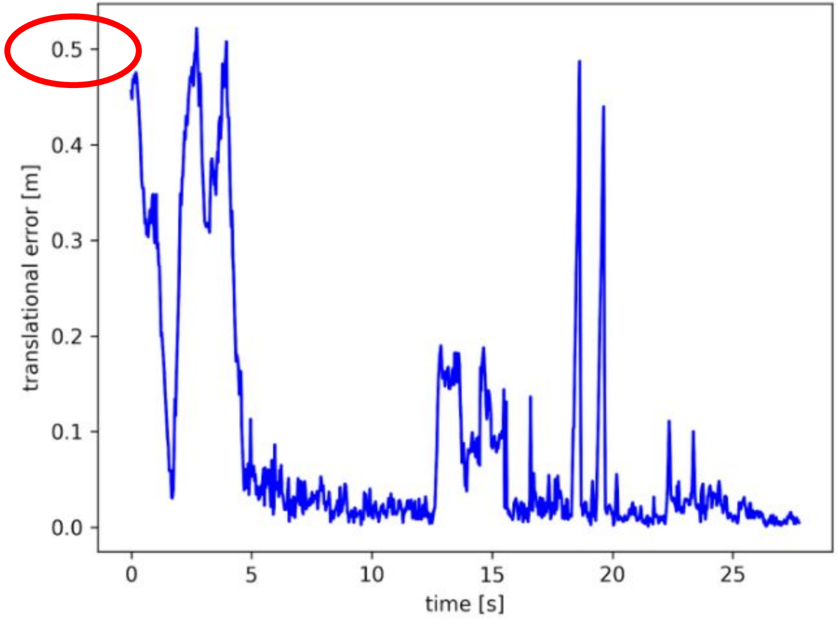
Figure 4.12: after masking

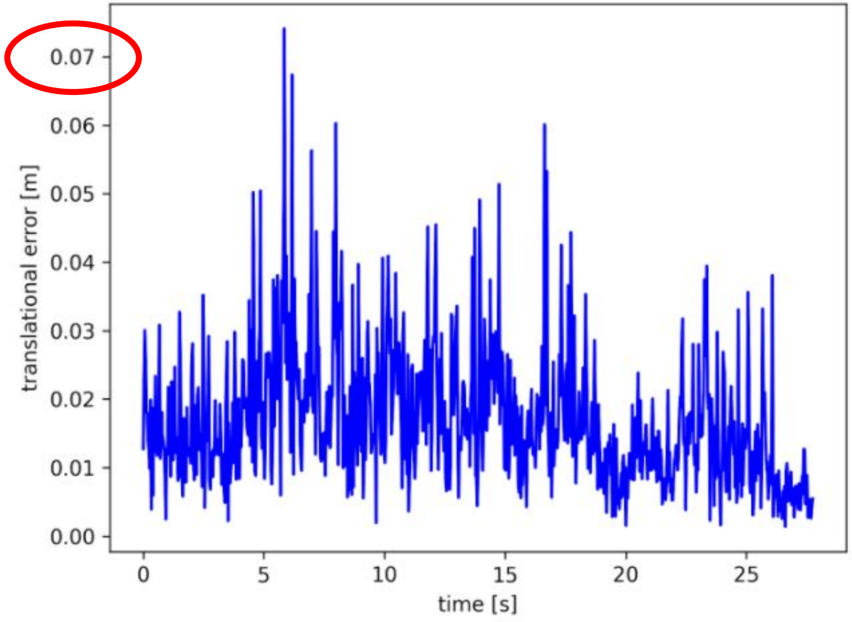
TUM walking_xyz (dynamic dataset) Reduction of tracking error





Relative translational error





Error Metrics

Without Mask

RMS error: 23.7222 cm
Relative Translational error: 16.69966 cm
Relative Rotational error: 3.093489 degree

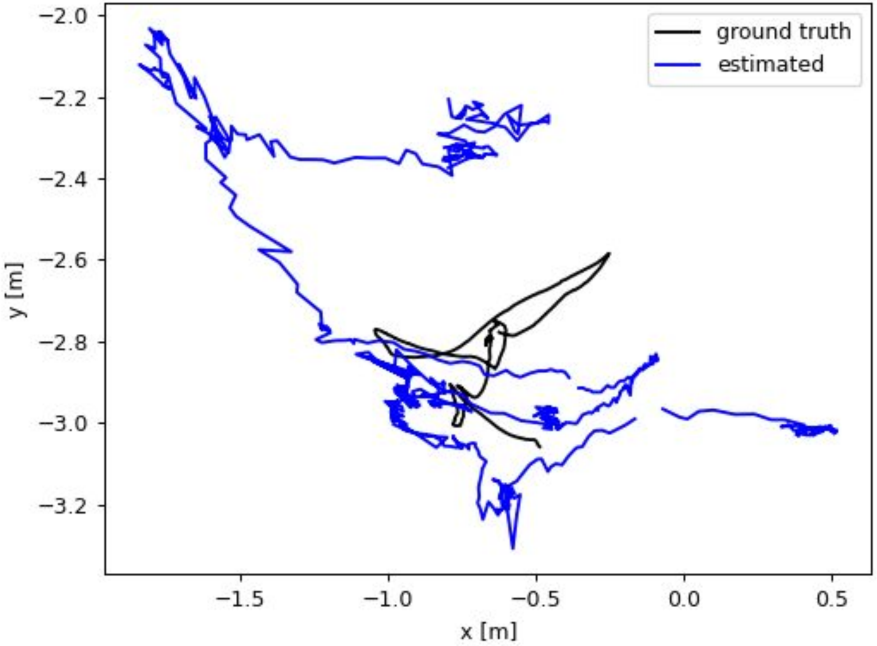
Best Case RMS error: 18.8568 cm

With Mask

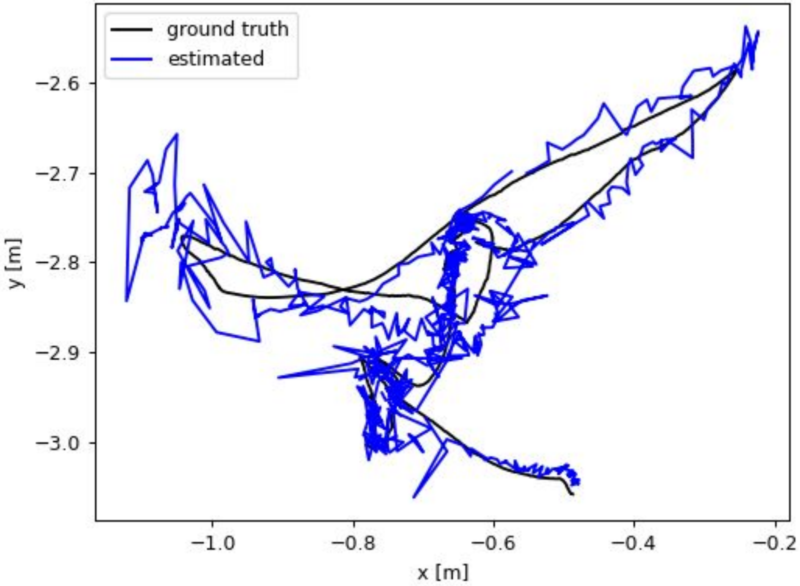
RMS error: 1.79716cm
Relative Translational error: 2.2598cm
Relative Rotational error: 0.6158846 degree

Best Case RMS error: 1.5409 cm

walking_rpy

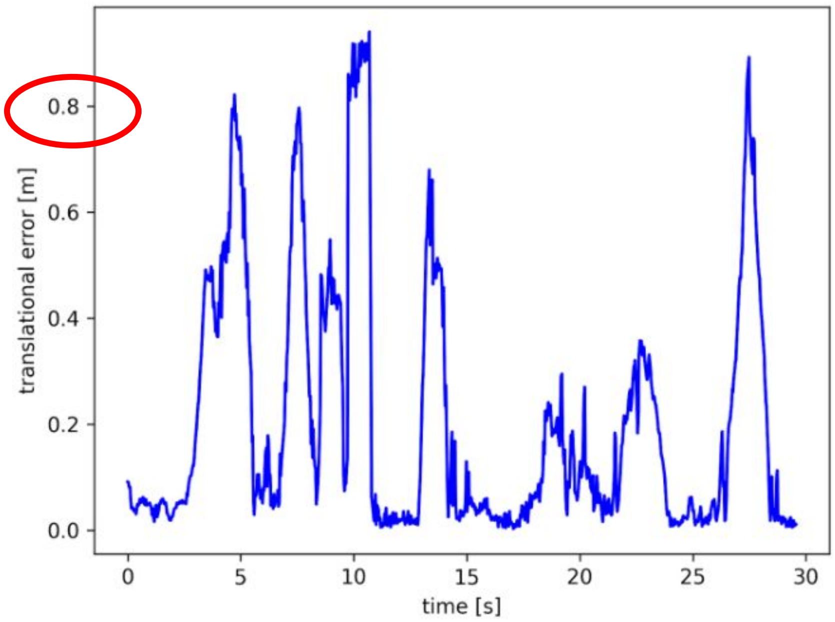


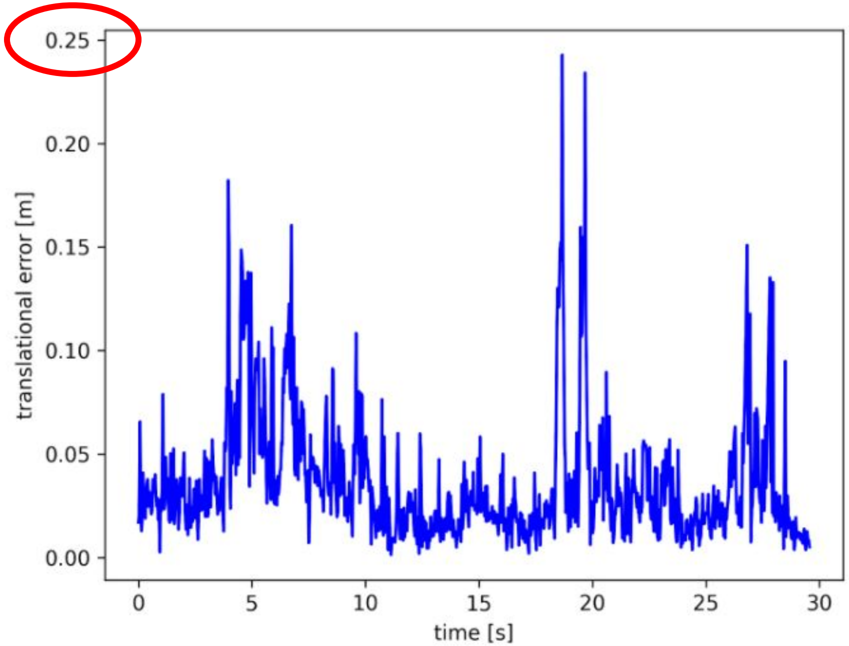
ground truth
estimated
−2.0
−2.2
−2.4
−2.6
−2.8
−3.0
−3.2
y [m]
−1.5
−1.0
−0.5
0.0
0.5
x [m]



ground truth
estimated
−2.6
−2.7
−2.8
−2.9
−3.0
y [m]
−1.0
−0.8
−0.6
−0.4
−0.2
x [m]

Relative translational error





Error Metrics

Without Mask

RMS error: 51.4982cm
Relative Translational error: 30.59184cm
Relative Rotational error: 6.0403042 degree

Best Case RMS error: 47.0009 cm

With Mask

RMS error: 3.9883 cm
Relative Translational error: 5.11032 cm
Relative Rotational error: 1.1446668degree

Best Case RMS error: 3.7272 cm

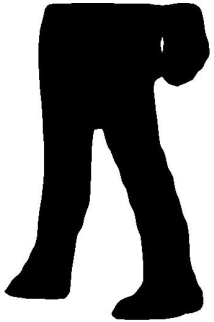
Let's compare Masks !!

Dataset & Methods	Validated on Locus Office Dataset		APSIS	
	mIOU(%)	FPS	mIOU (%)	FPS
ICNet	80.08	$\boxed{26.51525}$	83.69	$\boxed{20.90771}$
BiSeNetv1	84.09	13.71467	84.03	12.52348
DeepLabV3plus	88.77	7.28928	84.84	6.67264
UNetPlus	82.59	5.58920	84.34	7.57311
ICnet fine-tuned(ours)	83.27	24.03161	77.63	26.21884

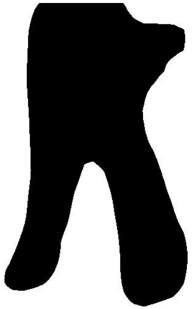
Table 5.3: Inference Speed mIOU Comparison of Segmentation Models

Note: All inference were carried out in *Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz (CPU only)*

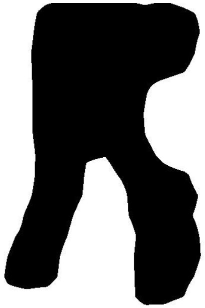
Model Comparison on MultiEnv dataset



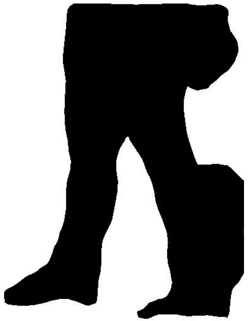
Ground Truth



ICNet Masking

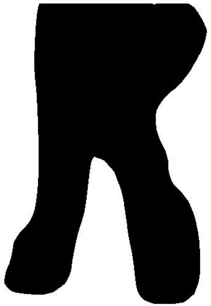


BiSeNet masking



DeepLabV3Plus Masking

UNet Masking



Our fined tuned Masking

Overlay Comparison of Masking Schemes

Original Image

ICNet overlay

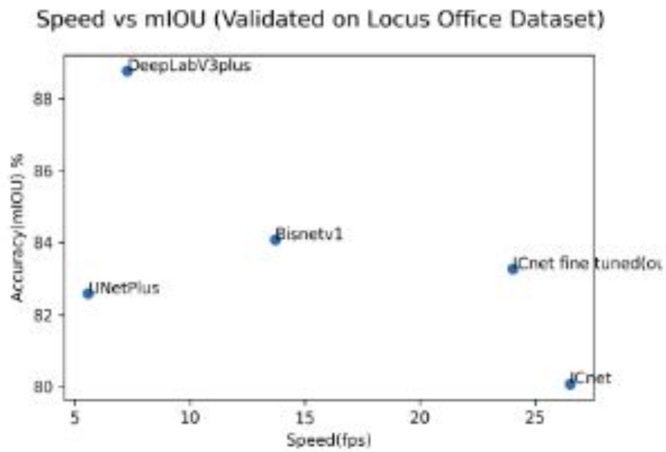
BiSeNet overlay

DeepLabV3Plus overlay

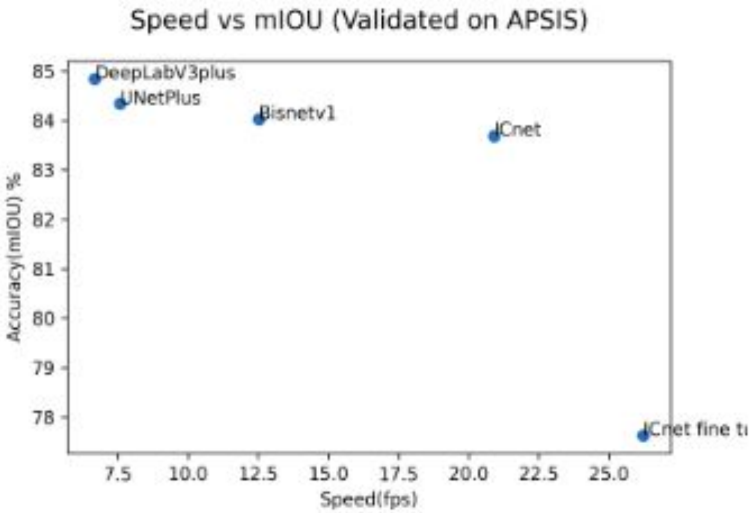
UNet overlay

Our Fined tuned overlay

Choose ICNet（speed over quality）



(a) Speed vs mIOU validated on Locus Office Dataset



(b) Speed vs mIOU validated on APSIS Dataset

Figure 5.16: Speed vs Accuracy Comparison of Models

Mask Generation Using ICNet

- ICNet for mask generation
  - Due to fastest inference speed in CPU
- Mask generated using pre-trained ICNet Model
- 3 branches model architecture
- Internally 320x320 resizing of input during inference

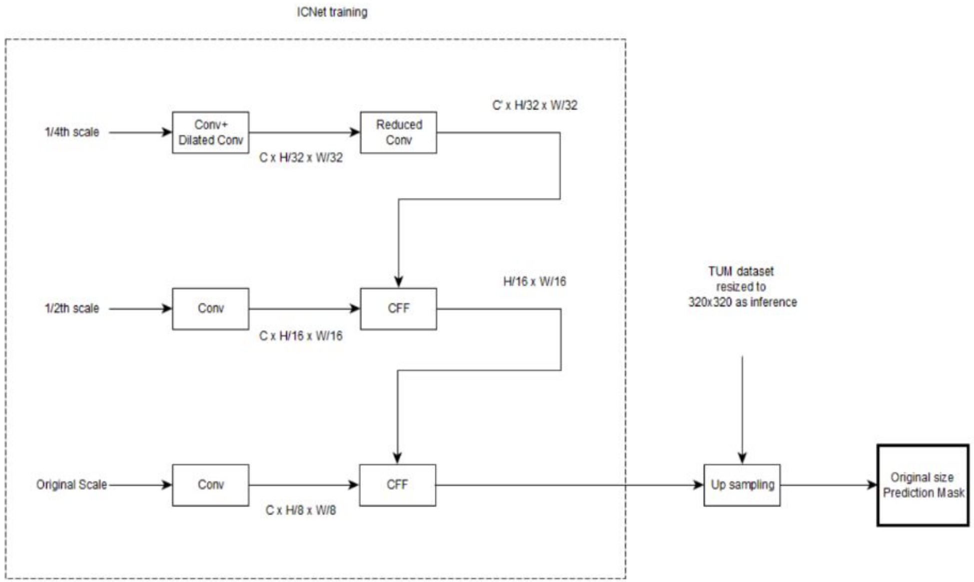
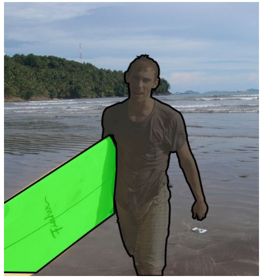


Figure 4.13: ICNet Inference

Further improvement of Mask

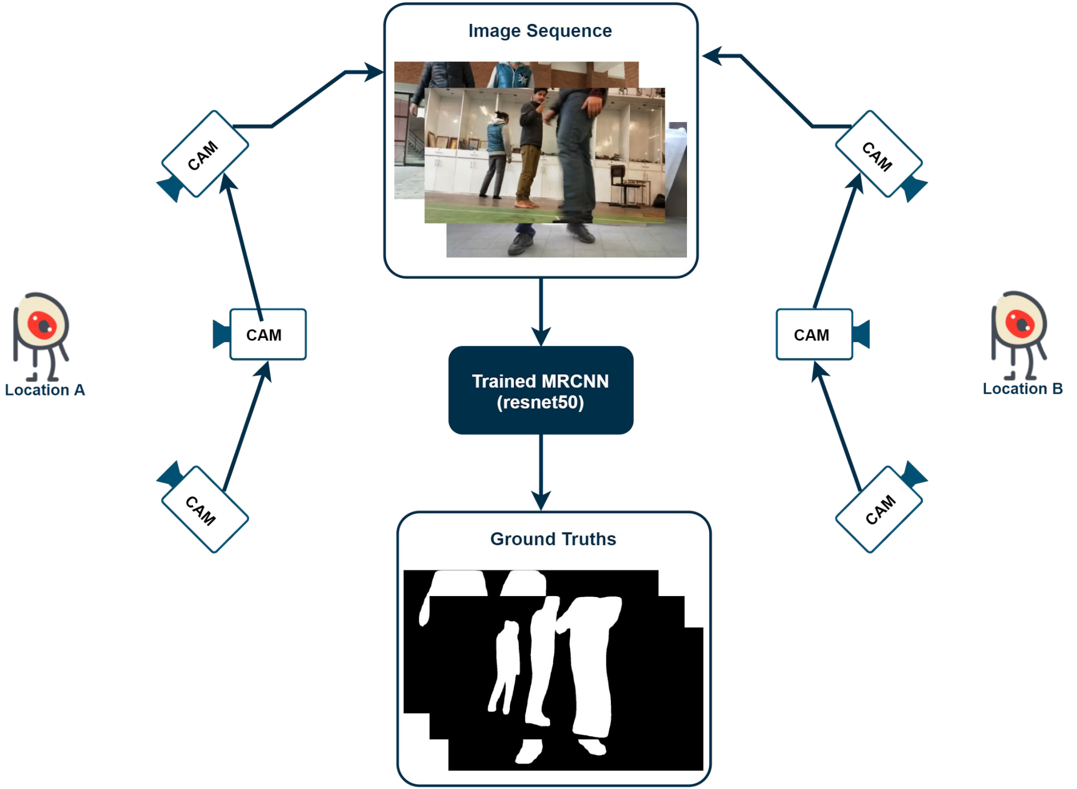


Robots perspective view

Common public human dataset

Focus on face and upper body

Custom Dataset Generation



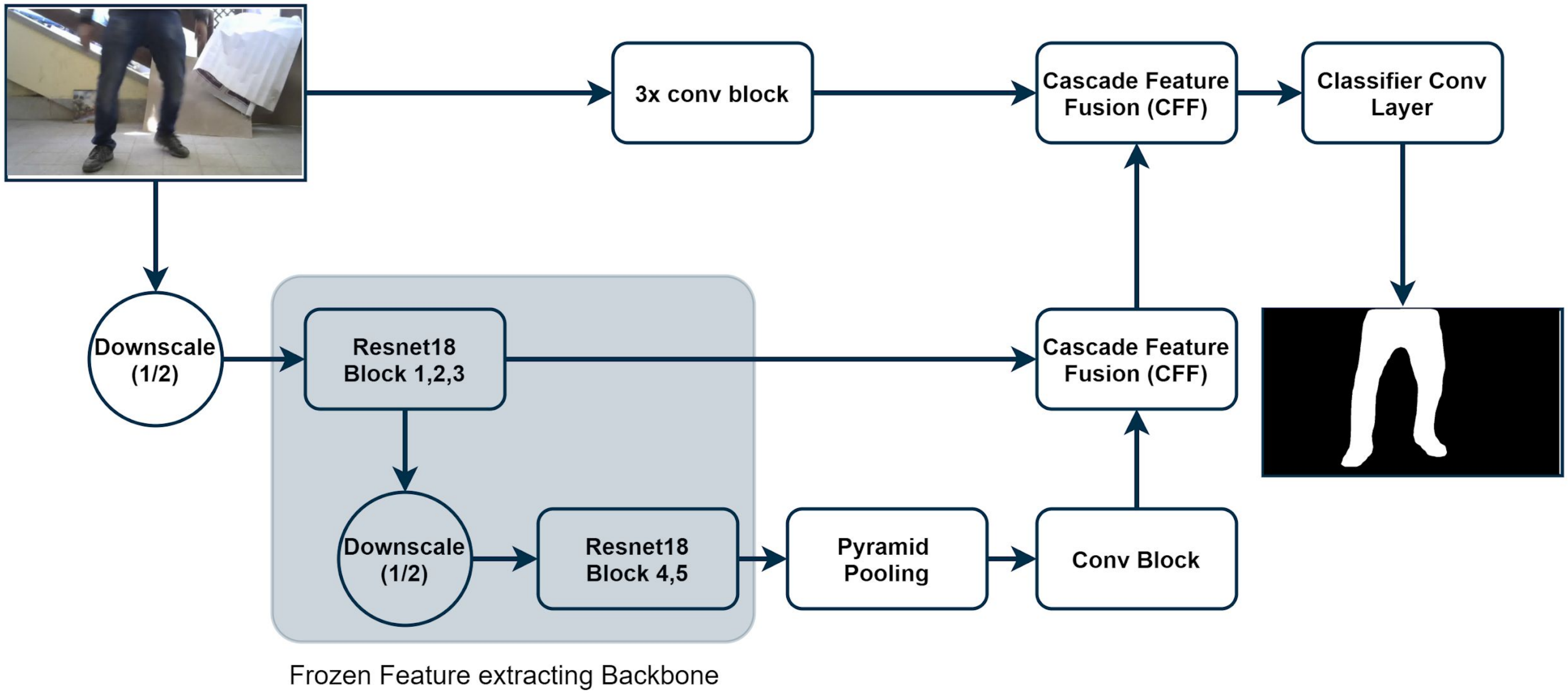
Multi Environment Walking Dataset (1435)

Taken as Training Set

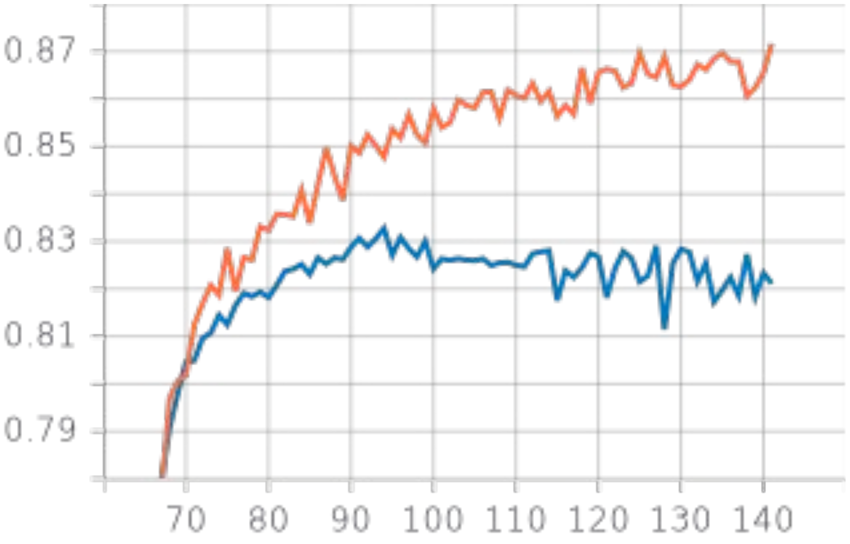
Locus Office walking dataset (1350)

Taken as Validation Set

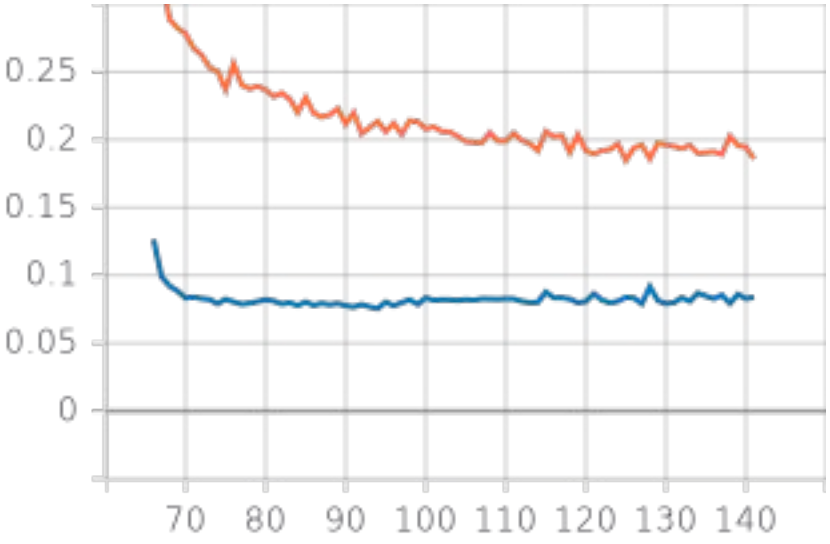
Fine Tuning ICnet



Fine Tuning ICnet

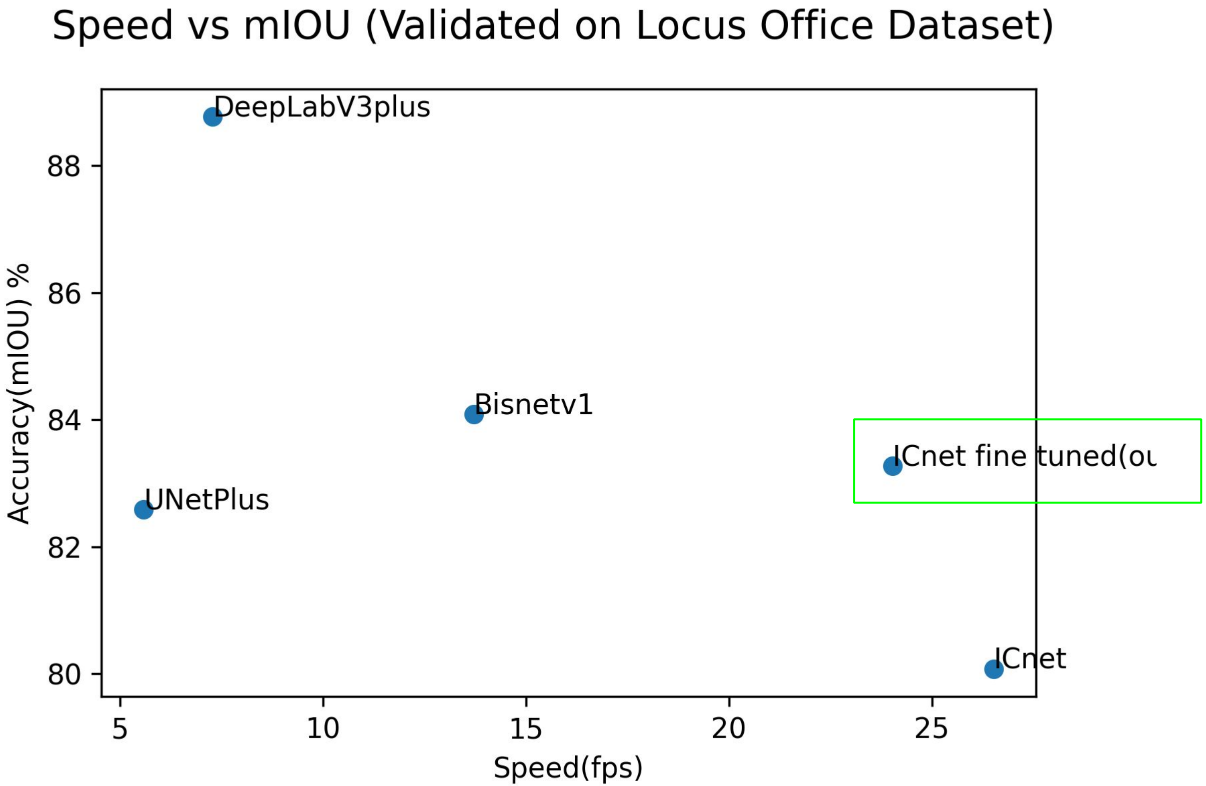


miou vs epoch



loss vs epoch

Fine Tuning ICnet



Limitation and further improvement

- Focused in **indoor** environment
- Considers **human** as main dynamic objects
- Could perform **motion segmentation** instead of semantic segmentation
- Make robot more robust to changes in **lighting**
- Improve performance in **texture** less environments

Thank you !!