Research Project
Computer Vision in Robotics

University of Stuttgart

Vinayak Patil

st174693@stud.uni-stuttgart.de

Institute of Architecture of Application Systems

Motivation

- It gives a robot to visually see and interact with its environment
- Robot vision expands computer vision approaches to meet the needs of robots and robotic systems.
- Navigation toward a specific target area while avoiding obstacles, finding a person and reacting to the person's directions, or detecting, recognizing, grasping, and delivering objects are all common tasks.
- Robot vision harness the power of visual sensing to observe and perceive the environment and react to it.

LAAS Research

Introduction

- Necessity of Vision in Robotics
- Flexibility in Functionality
  - Location, size and function
- Little Operator Oversight
  - Hardly any regular operator oversight is required
- Collaborative Work
  - Working with humans vision system provides more safety

IAS Research

Introduction

- Assigned Objectives
  - To interact with the robot's visual sensors and actuators.
  - Object detection in real time.

LAAS Research

Introduction

- Accomplished Objectives
  - ✔ ORB-SLAM is a Monocular SLAM using key-frames and features. It can close loops and conduct camera re-localization from a variety of views in real-time in vast landscapes.
  - ✔ YOLOv3 (You Only Look Once) is a cutting-edge, real-time object detecting technology. It is incredibly fast and precise.
  - ✔ Face Recognition - The world's simplest face recognition library allows you to recognize and manipulate faces from Python or the command line. Built with deep learning and dlib's state-of-the-art face recognition. The model is 99.38 percent accurate.
  - ✔ Visual sensors and actuators on the robot were used for interaction.

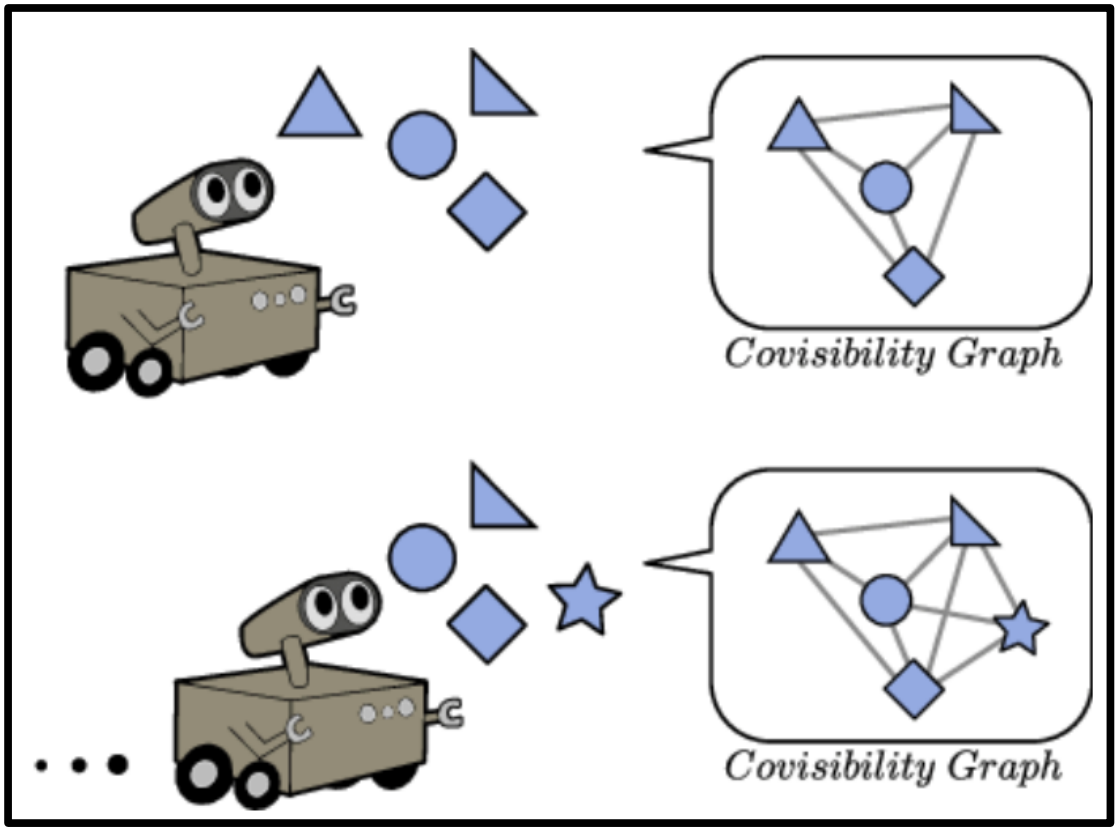
LAAS Research

Background: ORB_SLAM

- Bundle Adjustment: It estimates the 3D location of points in the environment with by camera with giving the pose of the camera.
- Features (map points): stores 3D position in world co-ordinate system.
- Key-frame: stores camera pose, camera intrinsics.
- Co-visibility graph: As the robot moves, it makes observations, detects landmarks, and creates a graph structure of which ones were observed together.
- Essential Graph: Contains all the nodes(key-frames) with less edges.
- Spanning Tree: A subset of the co-visibility graph's edges having a high co-visibility.

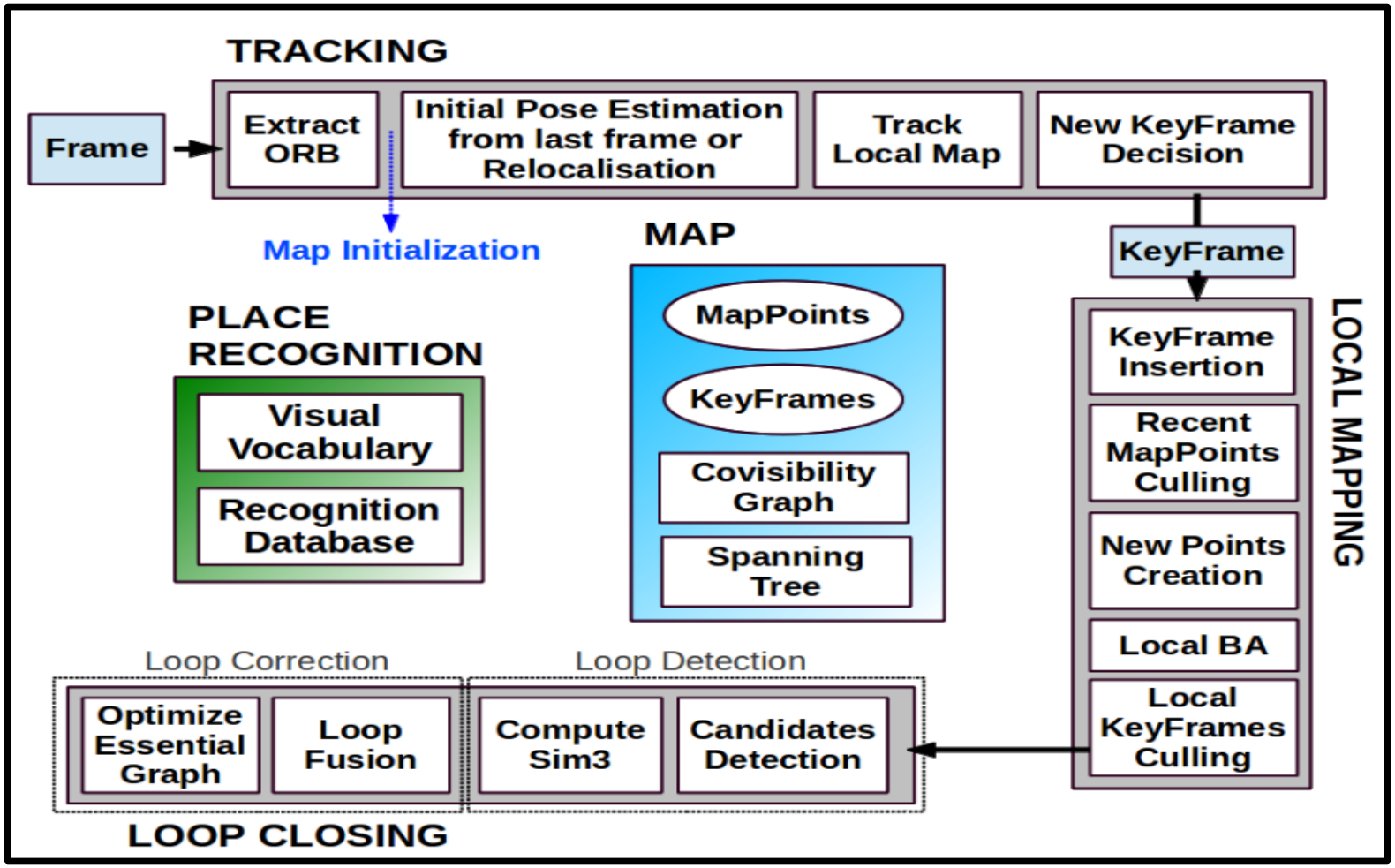
Background: ORB_SLAM

- Co-visibility Graph:



- The different shapes are the key-frames (nodes).
- A new key-frame is added to the Graph.

- Three Threads: Tracking, Local Mapping and Loop Closing.

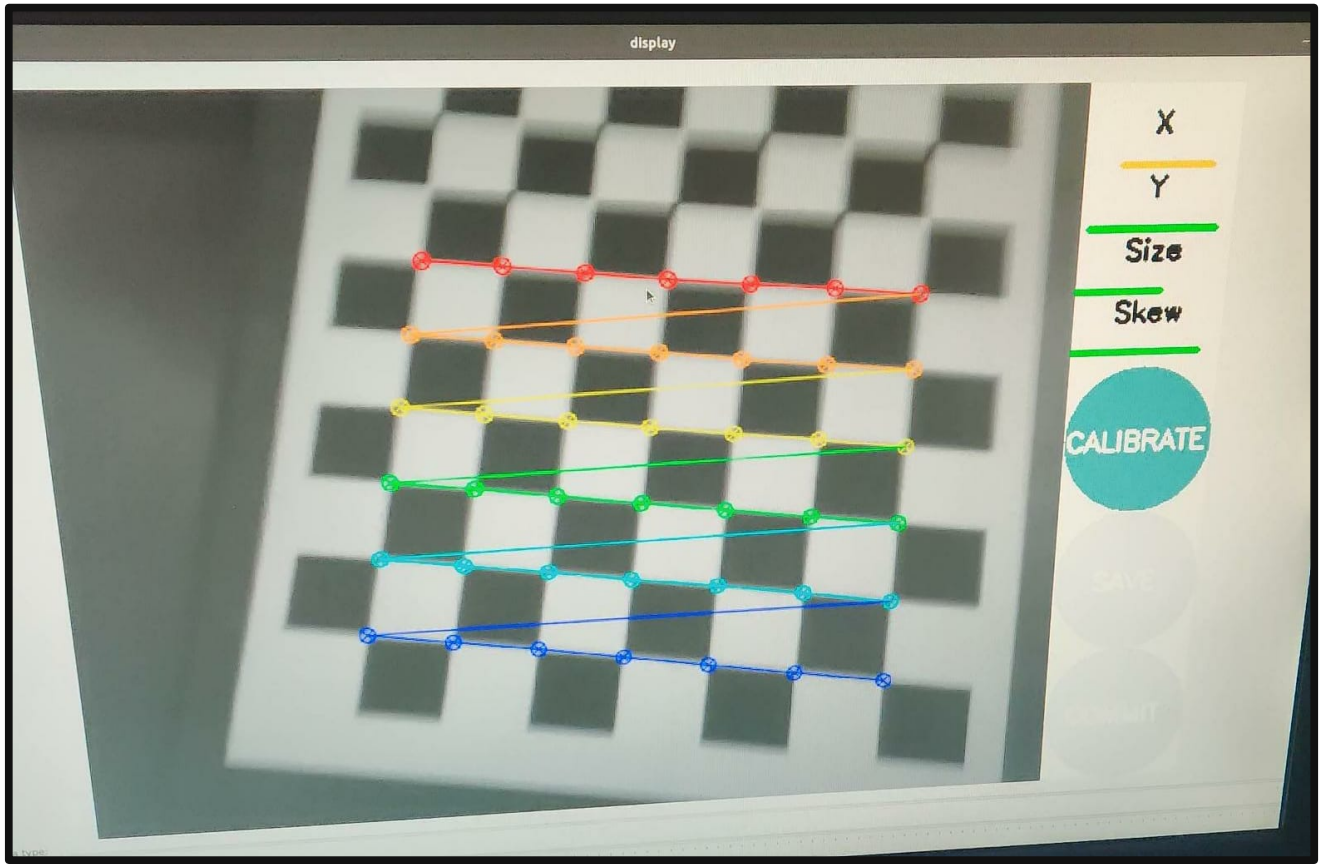


Evaluation: ORB_SLAM with USB_Camera.

- Camera Calibration: calculates the characteristics of an image or video camera's lens and image sensor. These characteristics can be used to adjust for lens distortion, measure the size of an object in world units, or find the camera's position in the picture. These tasks are used to recognize and measure objects in applications such as machine vision.
- Radial Distortion : k1, k2 and k3.
- Tangential Distortion : p1 and p2.
- Focal length : (fx, fy) , Optical centers : (cx, cy).

Extrinsic Parameters					Intrinsic parameters			
k1	k2	k3	p1	p2	fx	fy	cx	cy
0.0902	-0.143	0	0.0010	0.0002	640.29	640.44	292.01	242.62

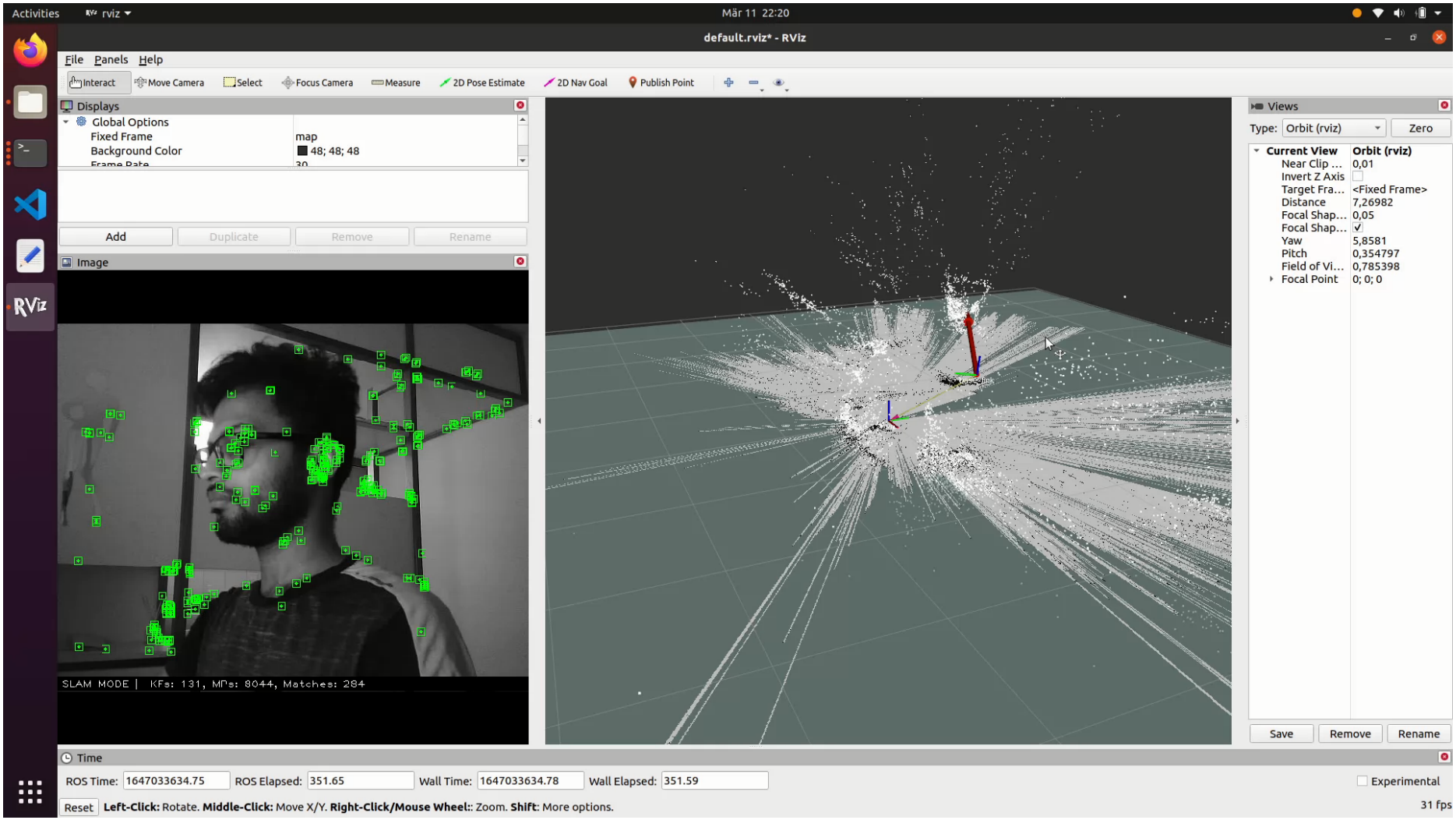
Evaluation: ORB_SLAM with USB_Camera.



- Monocular Pinhole camera.
- Square distance = 2.4cm

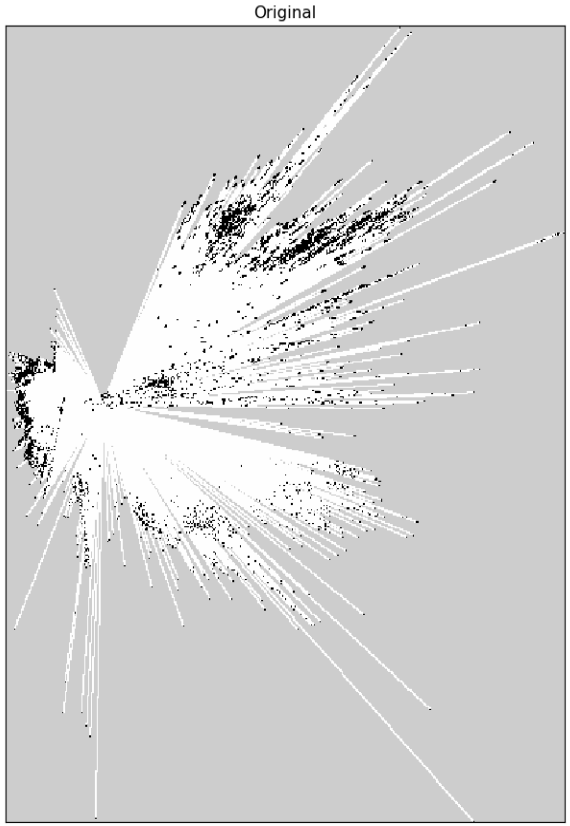
Evaluation: ORB_SLAM with USB_Camera.

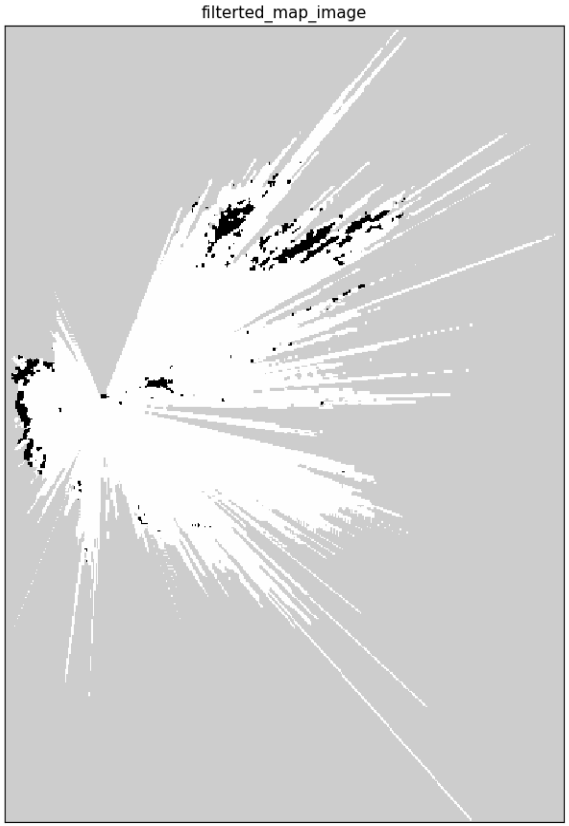
- Implementation video



Evaluation: ORB_SLAM with USB_Camera.

- octomap_server package used to get /Occupancygrid which is used by map_server to save the map in .pgm format and .yaml file.





- Median filter is used to smooth the map image.

LAAS Research

Evaluation: ORB_SLAM with Robot camera.

- Hurdles
  - 1. Not able to receive frames from the Robot camera.
  - 2. The color code of the received image was not RGB.
    - Cant implement Camera Calibration and ORB-SLAM.

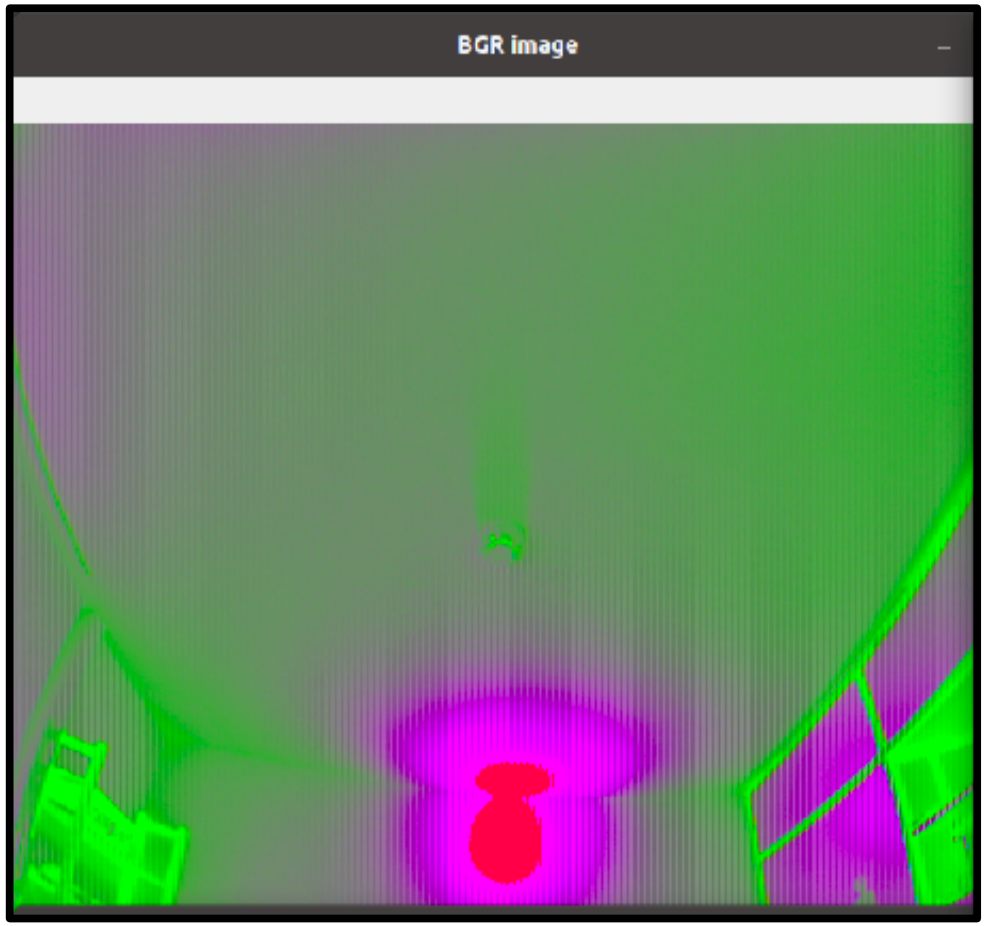
- Solution

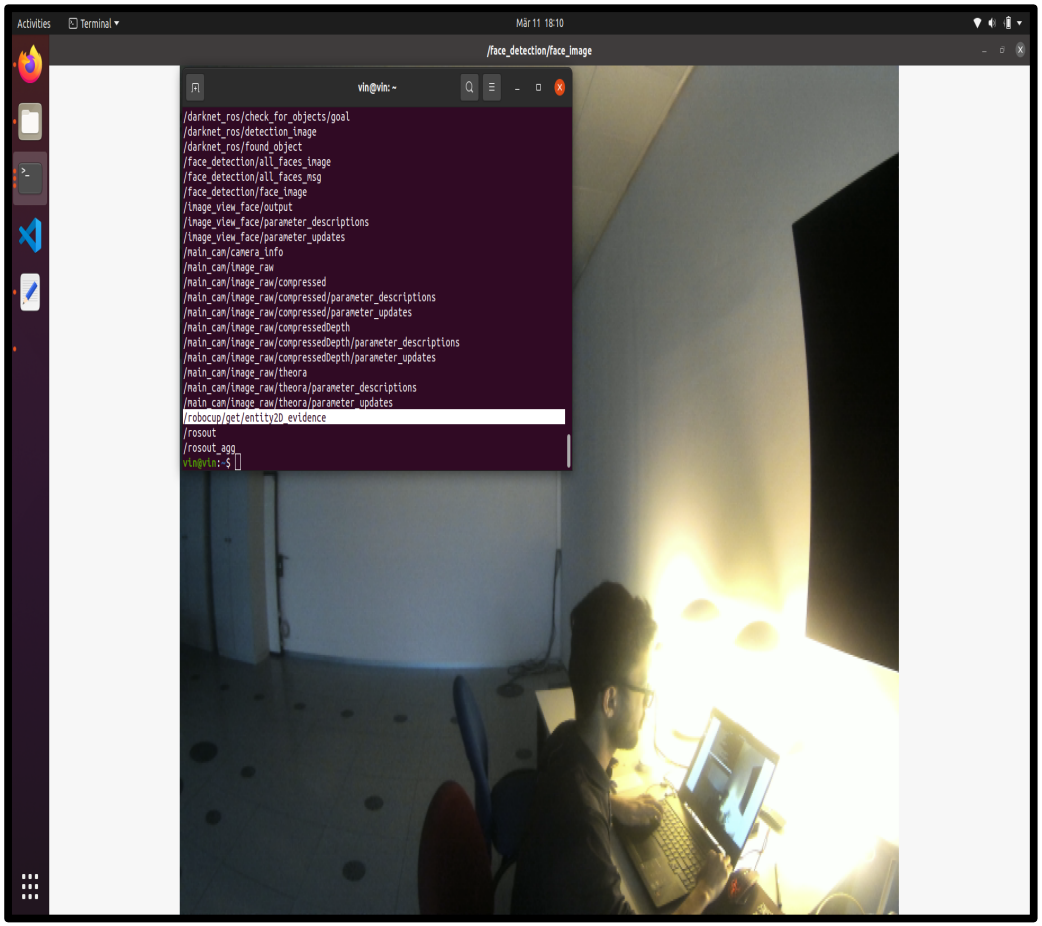
- docker run -it --privileged --network host -v /dev:/dev baoden/ohmni_rgbcam_ros:launch_ros bash
- docker run -it --network host --privileged -v /dev:/dev -e ROS_IP=192.168.178.104 baoden/ohmni_rgbcam_ros:launch_ros bash

HAS Research

Evaluation: ORB_SLAM with Robot camera.

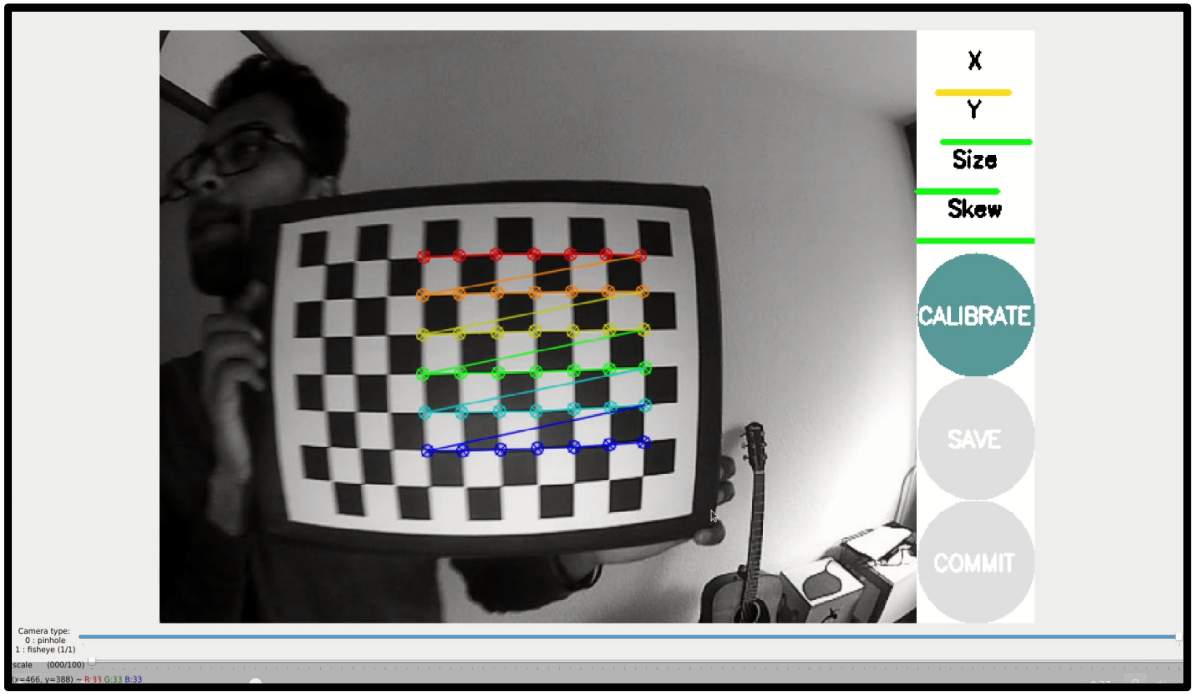
- In main_cam launch file changed the pixel format from uyvy to **yuyv.**





Evaluation: ORB_SLAM with Robot camera.

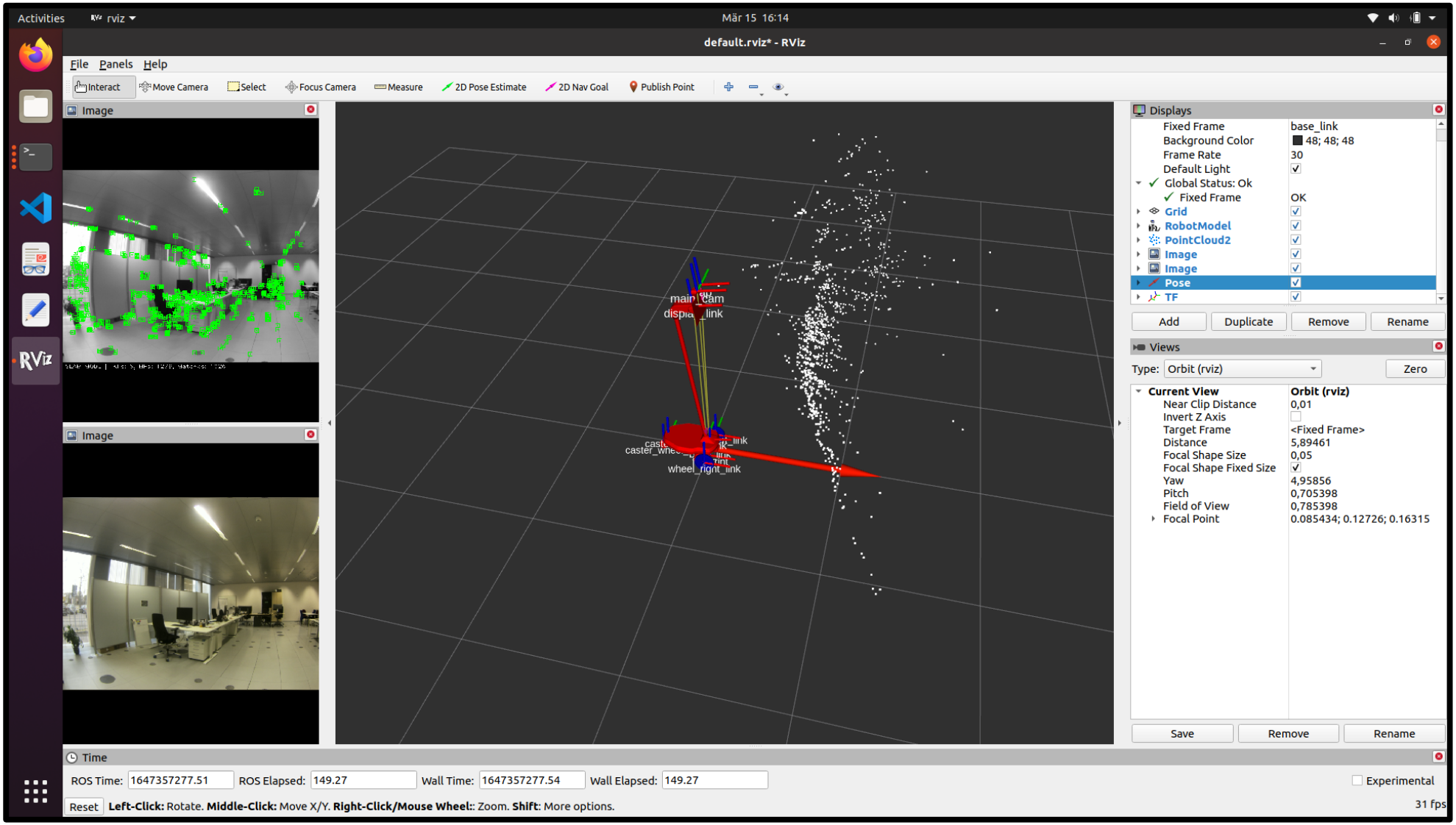
- Camera Calibration:



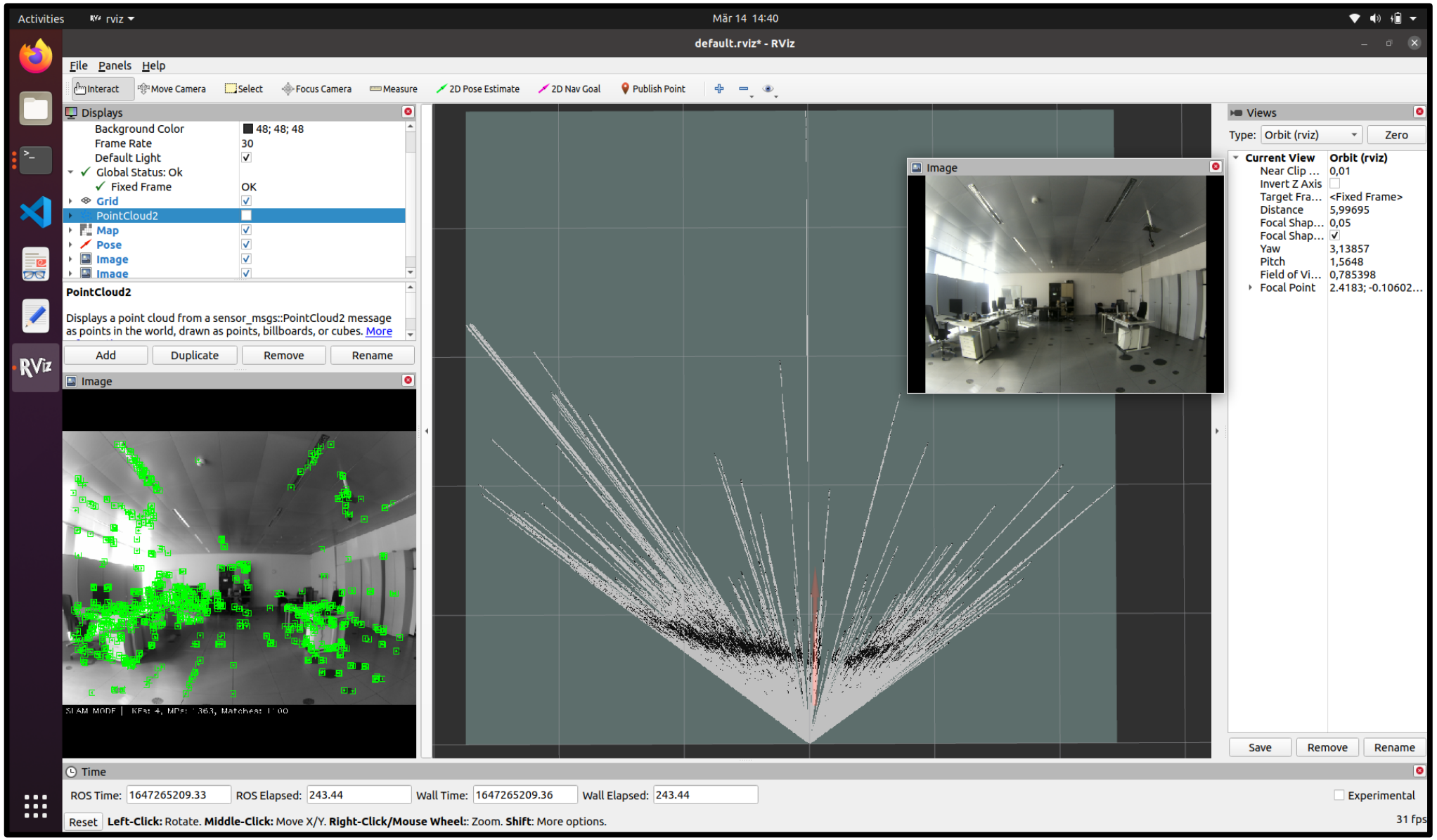
- Monocular Fish-eye camera.
- Square distance 2.4cm

Extrinsic Parameters					Intrinsic parameters			
k1	k2	k3	p1	p2	fx	fy	cx	cy
0.493	-0.273	0	-0.66	0.378	205.80	205	318.62	238.23

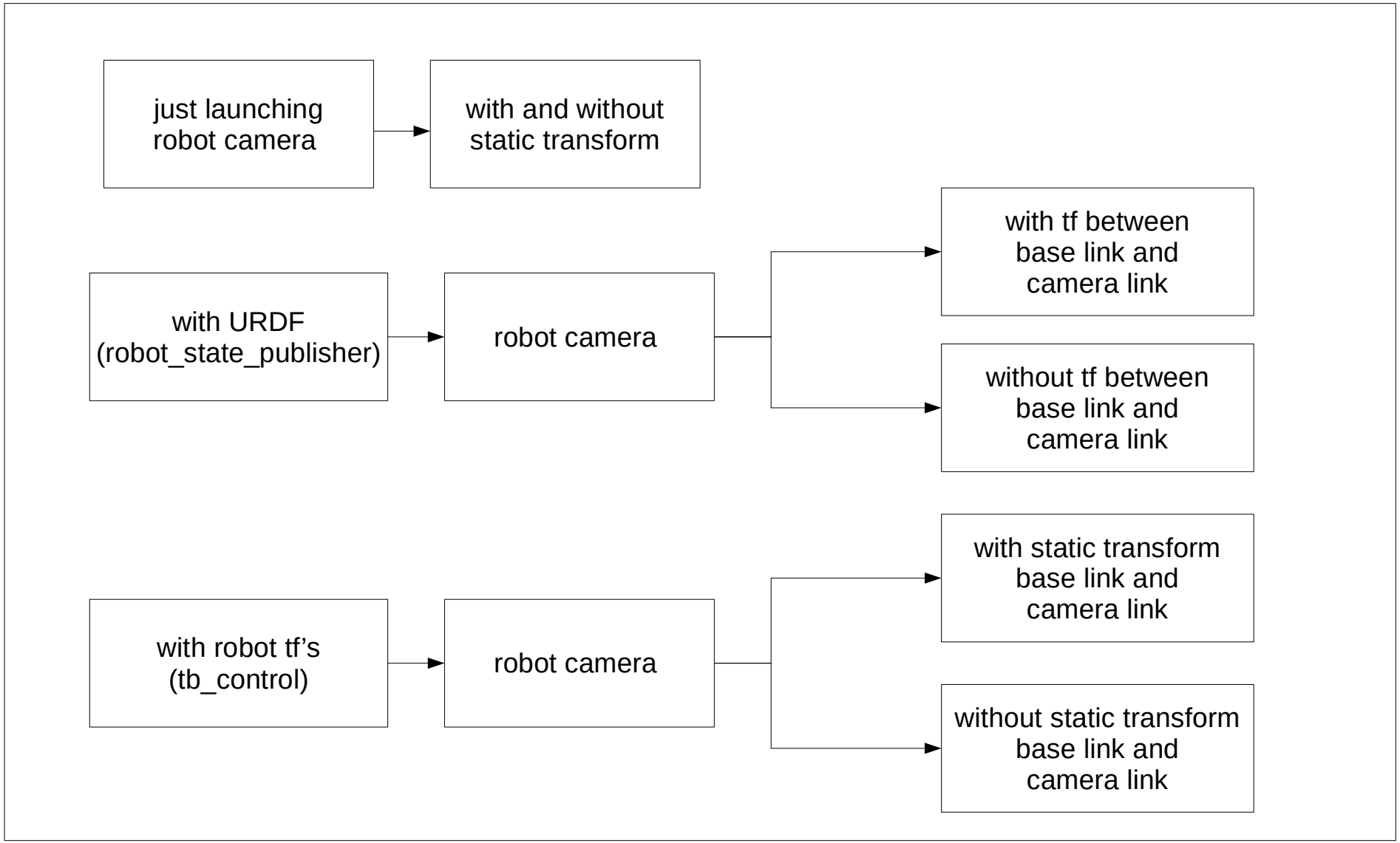
Evaluation: ORB_SLAM with URDF of Robot.



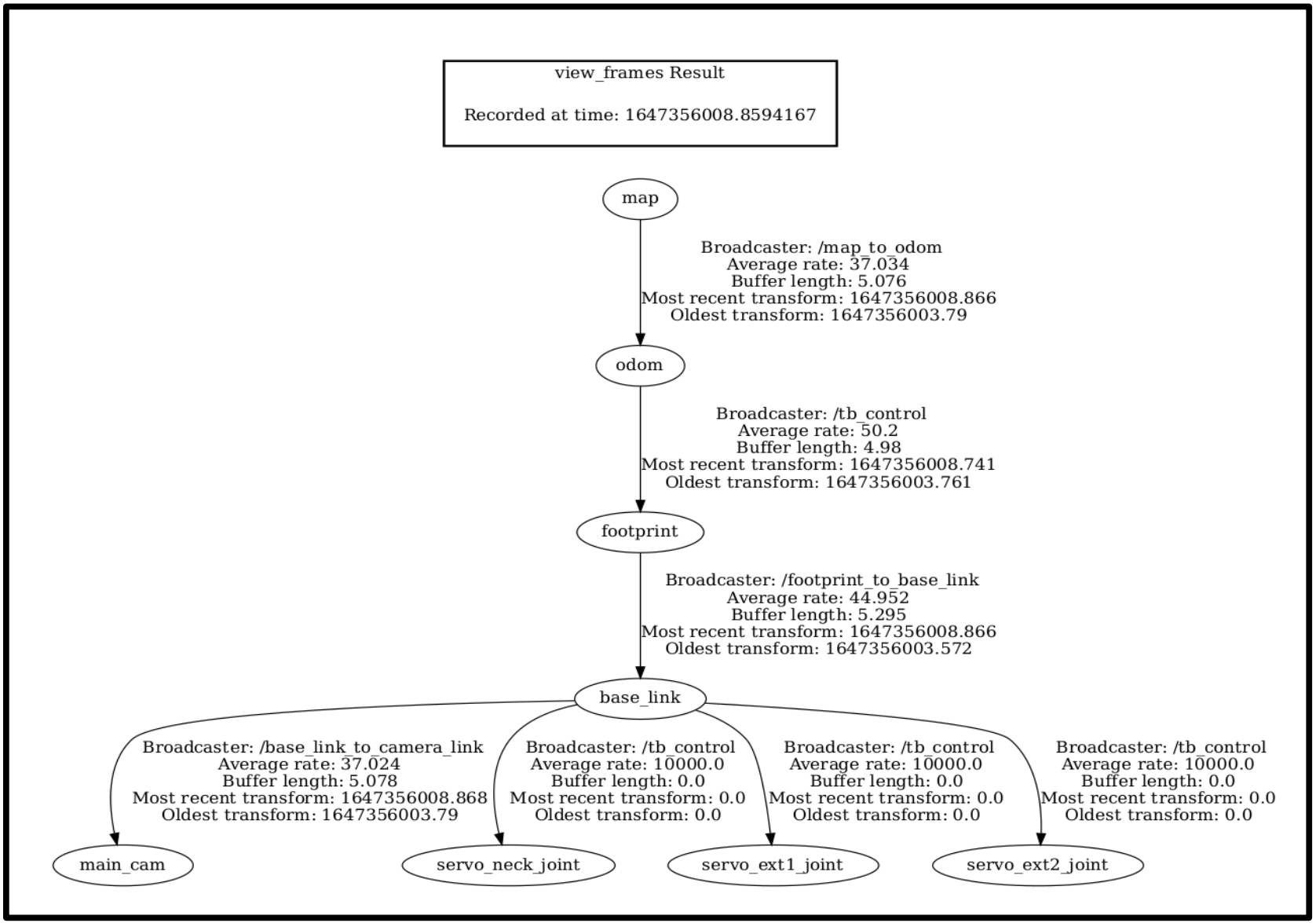
Evaluation: ORB_SLAM with Robot camera.



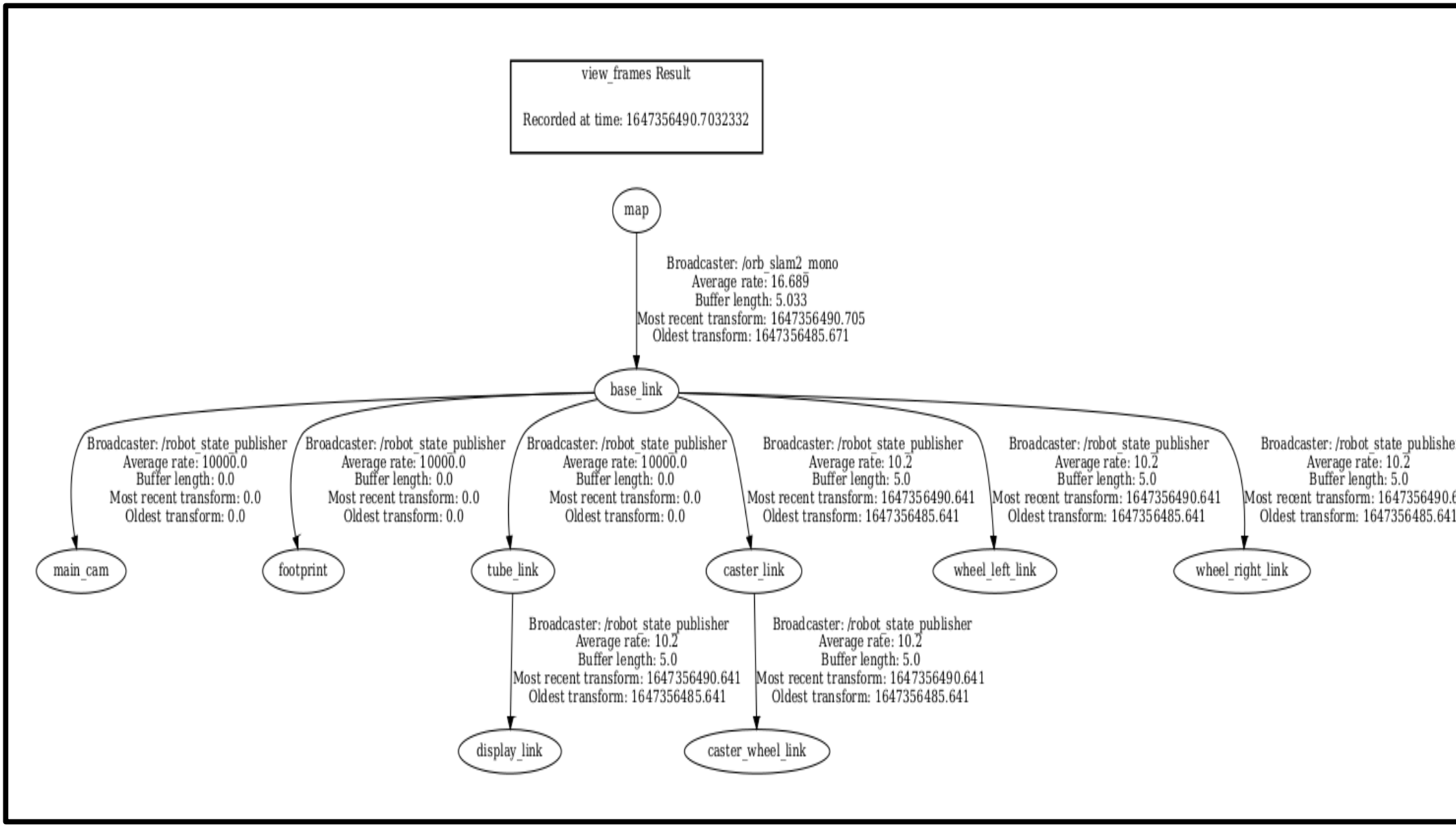
Evaluation: transform frame issue.



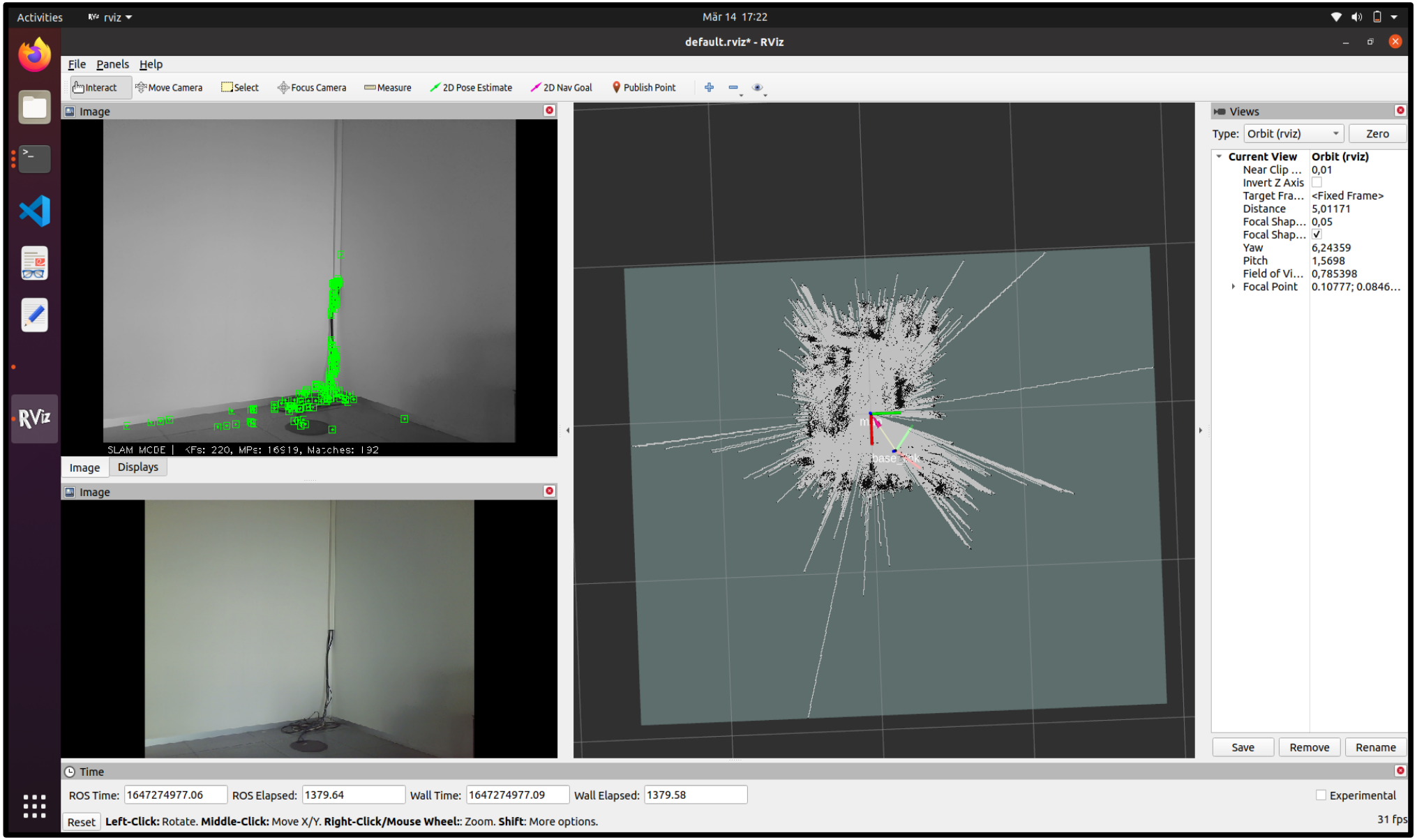
Evaluation: ORB_SLAM.



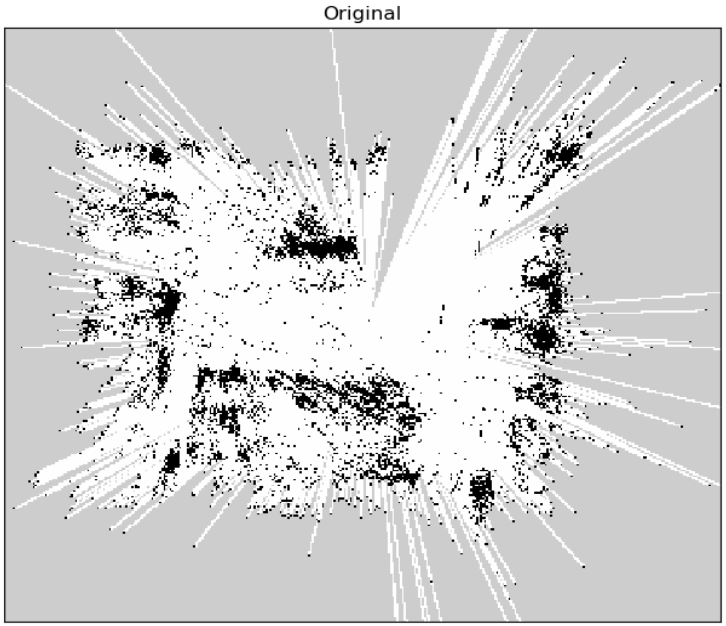
Evaluation: ORB_SLAM.

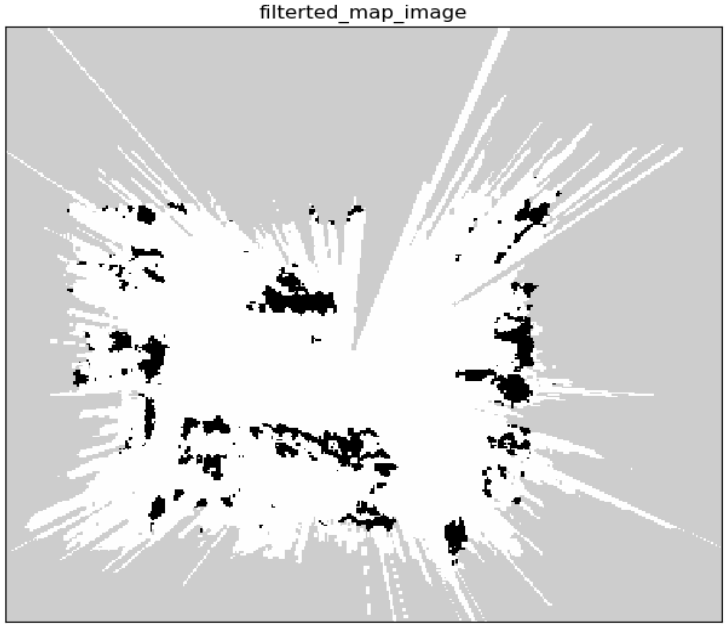


Evaluation: ORB_SLAM with USB camera.

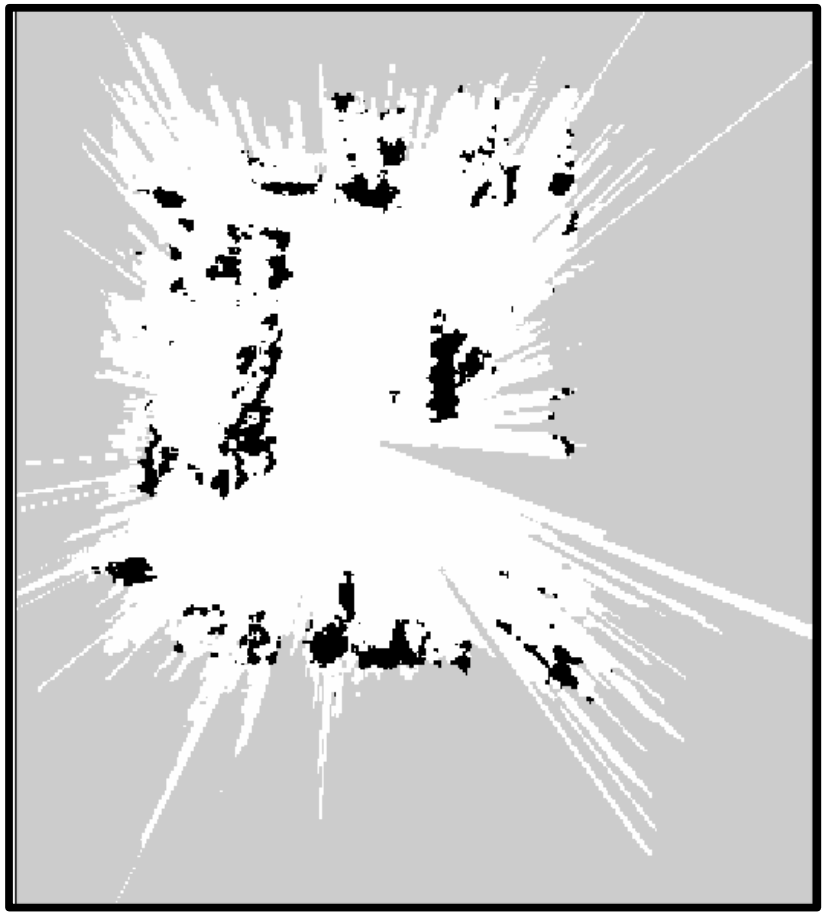


Evaluation: ORB_SLAM with USB camera.

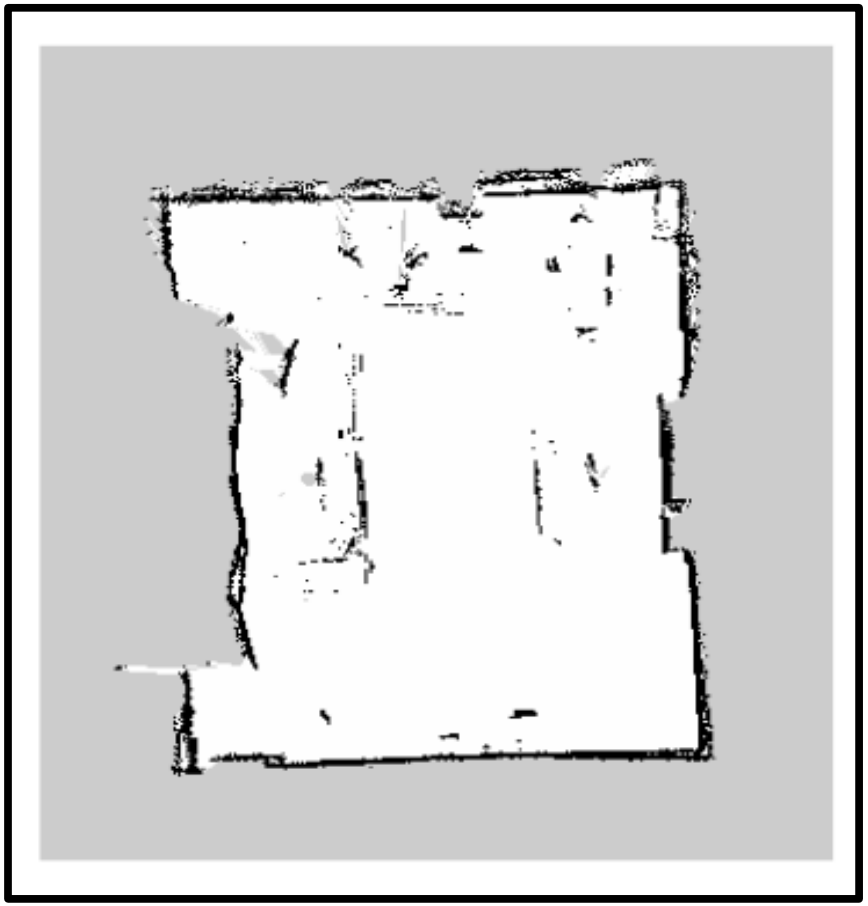




Evaluation: Comparision.



Map with ORB-SLAM
with 10Euro camera



Map with Lidar SLAM
with 105Euro YD Lidar

Approach: Object detection with YOLOv3.

- A pre trained model is used to detect the objects from the camera frame.
- It uses a single neural network to process the entire image.
- The image is divided into regions by this network, which predicts bounding boxes and probabilities for each region.
- The projected probabilities are used to weight these bounding boxes.
- COCO data-set : It is a large-scale object detection, segmentation, and captioning data set having 80 object categories.
- Configuration file: It contains the hyper-parameters and neural network structure (CNN).
- Weights and Biases file: The information regarding weights and biases used by the neural network is stored.

LAAS Research

Approach: Object detection with YOLOv3.

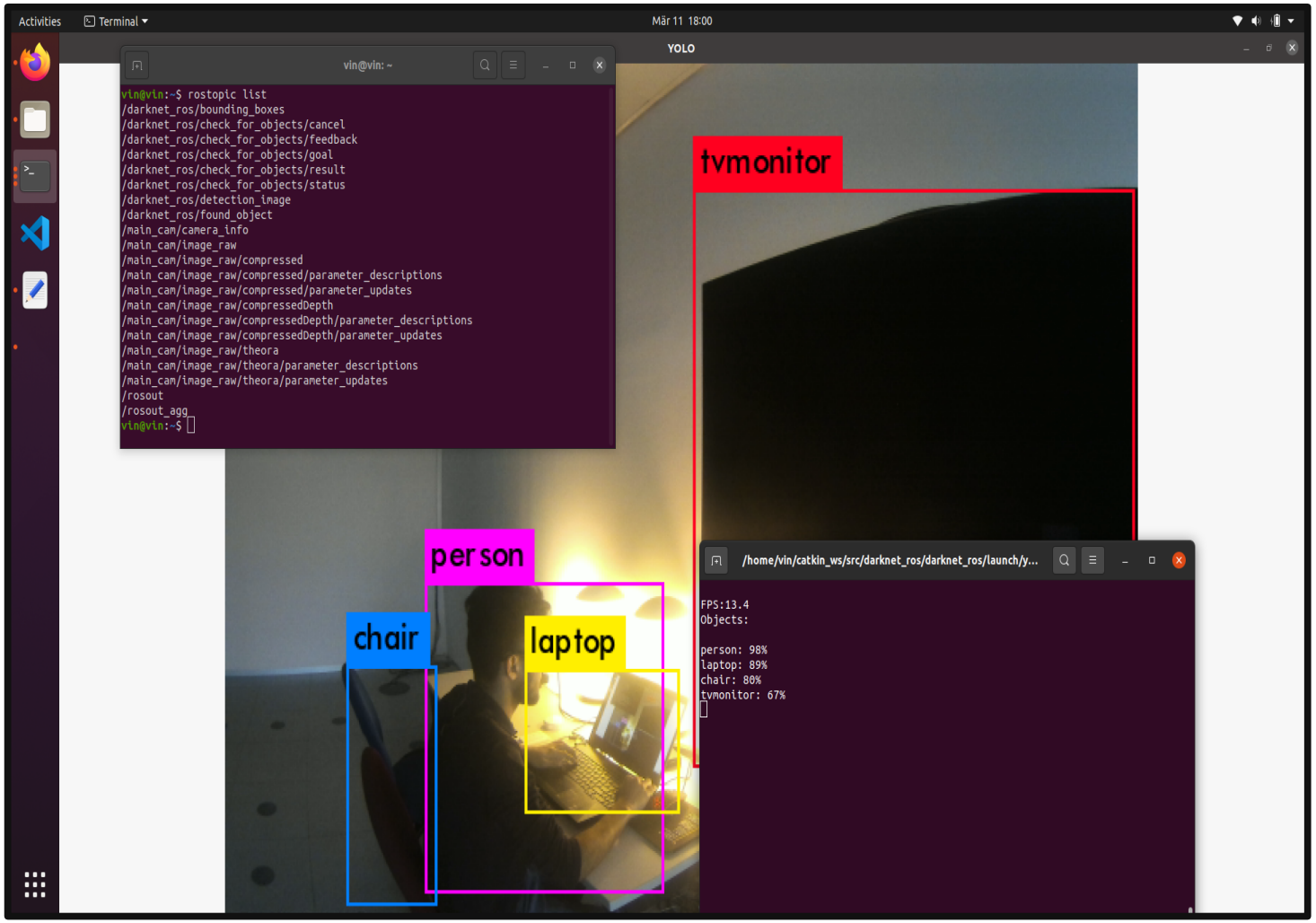
- The neural network uses 53 convolution layers for processing the image and detecting the object with bounding box with its probability. Kernel size used is 3*3 and 1*1.

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

- The number of feature map is increased in parallel to the reduction of their spatial size to avoid too much information loss.

Evaluation: Object detection with YOLOv3.

- The algorithm requires GPU for its processing, so CUDA is installed on local system.



- Object with its probabili is displayed.
- ROS topics regarding object detection are published.

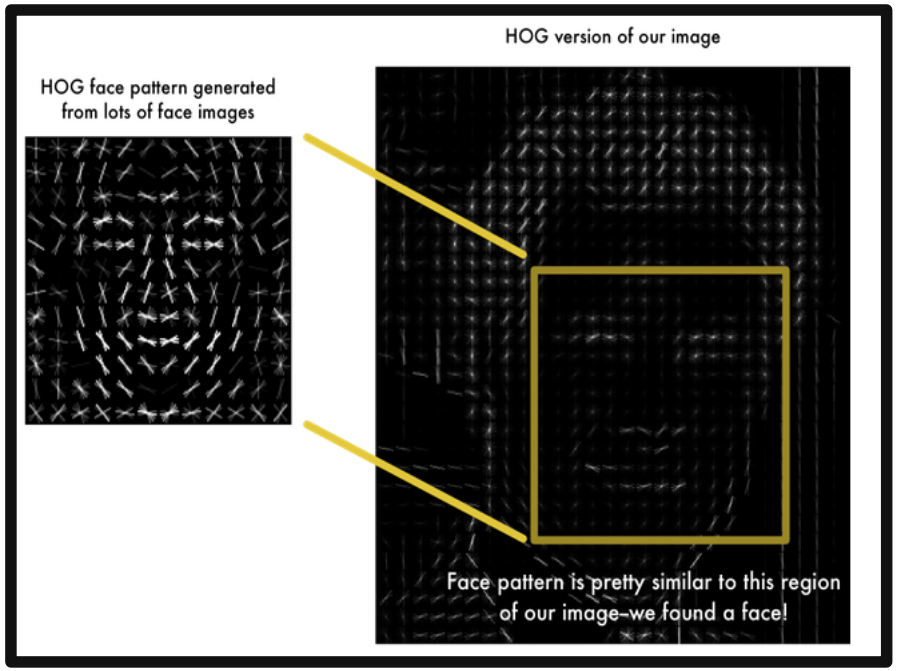
Evaluation: Object detection with YOLOv3.

- We can train our own data-set with same neural network structure of YOLOv3
- It gives the weight file, which can be used for custom object detection.
- https://github.com/vin3697/Custom_Obeject_detection-My-Guitar
- When tiny weight and configuration file is used, it reduces the processing power but decreases the accuracy as well.
- It uses less number of convolution layer that is 13 layers only which affects the accuracy.

IAAS Research

Approach: Face recognition with dlib library.

- Face detection with Histogram of Oriented Gradients.
  - Grey scale image is enough to detect the faces in the image.
  - Independent of the brightness of the image.
  - Gradient : it shows the flow from light to dark across the entire image depending upon the surround pixels for each pixel.

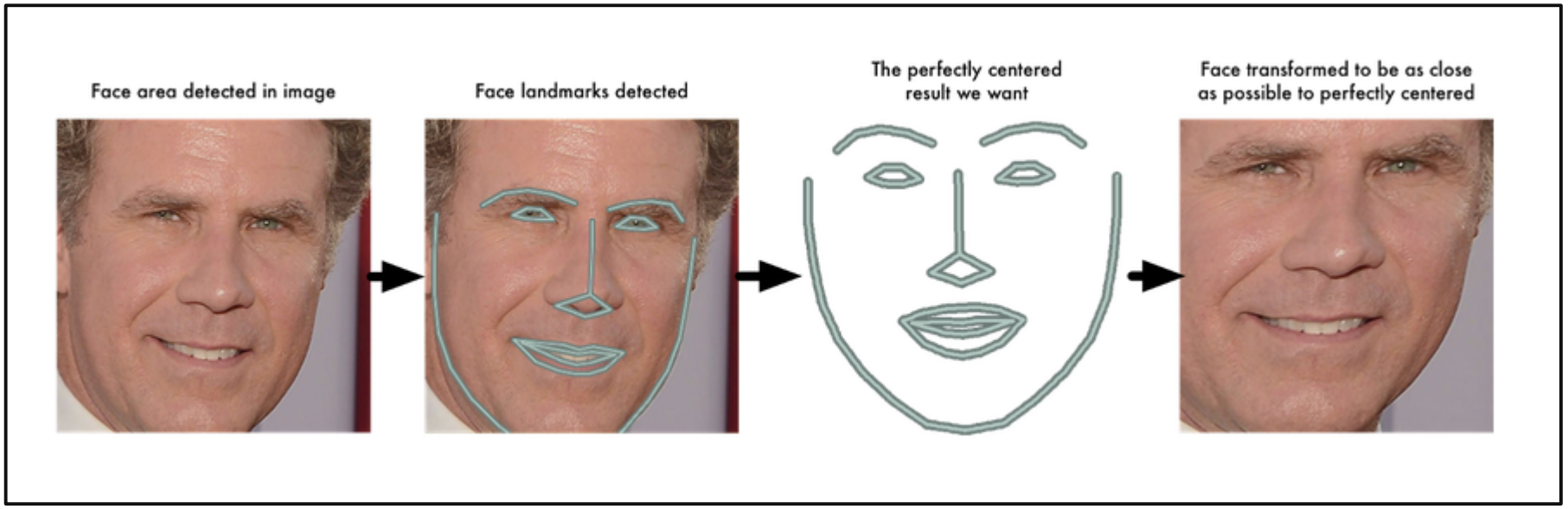


- All we have to do to locate faces in this HOG image is find the region of our image that looks the most like a known HOG pattern that was taken from a bunch of other training faces.

HAS Research

Approach: Face recognition.

- Face landmark estimation
  - The main concept is that we will identify 68 distinct places (known as landmarks) on every face, such as the top of the chin, the outside edge of each eye, the inside edge of each brow, and so on.
  - Then, on any face, we'll train a machine learning algorithm to locate these 68 specific points



Approach: Face recognition.

- Encoding Faces:
  - Run the centered face image through a neural network that understands how to measure facial traits.
  - Keep track of the 128 measurements.
- Look through all of the faces we've already measured to see who has the most similar measurements to ours.

IAAS Research

Evaluation: Face recognition.

- In run-time, it labels the faces with ID.
- Label these faces with the corresponding person name.
- It hardly takes 2 minutes to do label and detect the person face in run time.

IAS Research

Conclusion

- Learned ROS to implement the actual ideas into robot environment.
- Learned Docker and Android Debug Bridge(ADB).
- Receiving the image frame from robot was the most challenging task.
- Successfully interacted with camera and actuators of the robot with help of programs (python and c++).
- Implementation on ORB- SLAM (VSLAM) and creating a 2D map of the environment.
- Object detection with help of YOLOv3 algorithm.
- Face recognition with help of dlib library.
- Applying static transform (tf) raises a issue.

LAAS Research

Conclusion

- Latency for Object detection and Face recognition.

	Number of samples	Latency/Lagging (min-max)	Average value
Object detection	100	690ms-780ms	731ms
Face recognition	100	850ms-940ms	886ms

LAAS Research

Outlook

- Building Navigation Stack for ORBSLAM.
- Higher YOLO versions can be implemented to achieve more accuracy in the results.
- Depth camera can be used externally to implement object detection and collision avoidance.
- RTAB-Map (Real-Time Appearance-Based Mapping) another VSLAM algorithm can be implemented with RGB-D camera and Lidar.
- With help of Rpi different sensor and actuators can be embedded with robot to give it more features. (it solely depends on application)
- Using mic to interact with the robot functionary. (NLP)

IAS Research

References

- file:///tmp/mozilla_vin0/Probabilistic_Place_Recognition_with_Covisibility_.pdf
- https://www.mathworks.com/help/vision/ug/camera-calibration.html
- https://docs.opencv.org/3.4/dc/dbb/tutorial_py_calibration.html
- https://pjreddie.com/media/files/papers/YOLOv3.pdf
- https://cocodataset.org/#home
- https://pypi.org/project/face-recognition/
- https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78

IAS Research

Acknowledgements

Thank you very much to Prof. Dr. Marco Aiello and Mr. Nasiru Aboki.