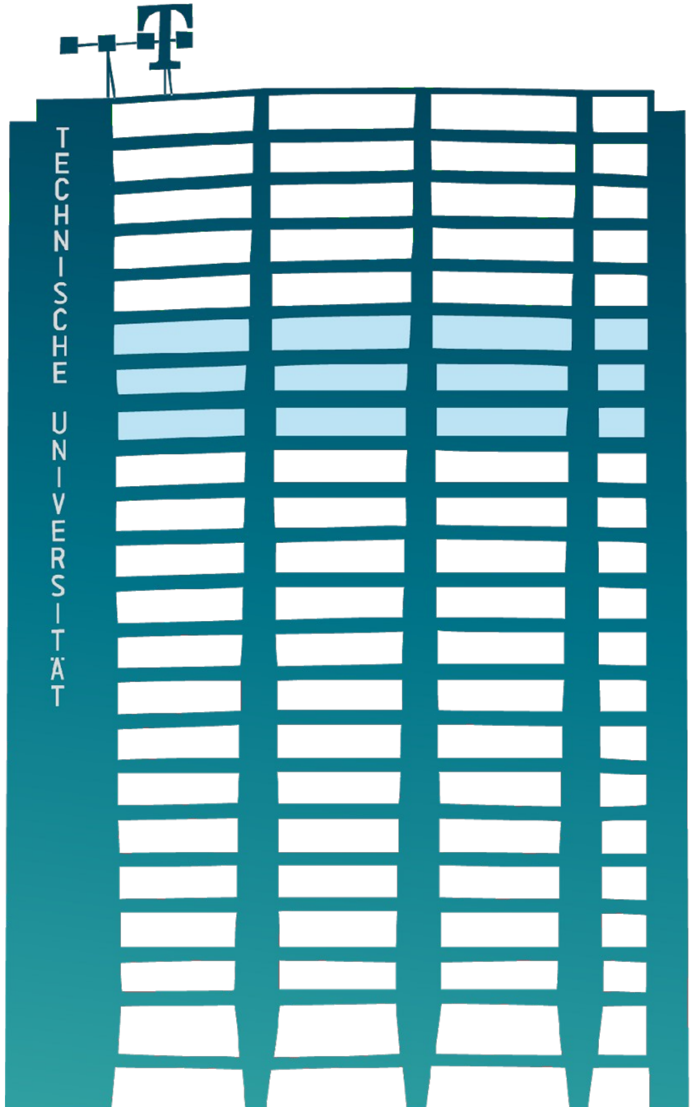


Applications of Robotics and Autonomous Systems (APP-RAS)

Project Course Module: 40305, 40889

Lecturer: Prof. Dr. Sahin Albayrak, Dr. Yuan Xu

Theme: Autonomous Driving

Supervisors : Yuan Xu, Yuchen Liu, Evgeny Gorelik, Philipp Grosenick

start time: 14:15

Contact: yuan.xu@dai-labor.de

Implementation of Visual SLAM on Bearcar

Yuan, Yuchen

Introduction

Visual SLAM (Simultaneous Localization And Mapping) provides a method for visually estimating the position of a robot relative to its start position, known as VO (visual odometry). It could also integrate IMU (Inertial Measurement Unit) data, which include accelerometers and gyroscopes. VSLAM can track the motion of key points between the two consecutive images to estimate the motion of the camera, which is then used to compute odometry.

Project Platform

Bearcar: A 1/10 scale autonomous vehicle equipped with:

Nvidia Jetson AGX Xavier (embedded computer)

Intel Realsense D455, which is a RGBD camera with IMU.

2D LiDAR

ESC (Electronic Speed Control) motor controller

Software: Autonomous driving software stack with an available codebase.

Source: https://media.githubusercontent.com/media/NVIDIA-ISAAC-ROS/.github/main/resources/isaac_ros_docs/repositories_and_packages/isaac_ros_visual_slam/cuvslam_ros_3.gif/

Implementation of Visual SLAM on Bearcar

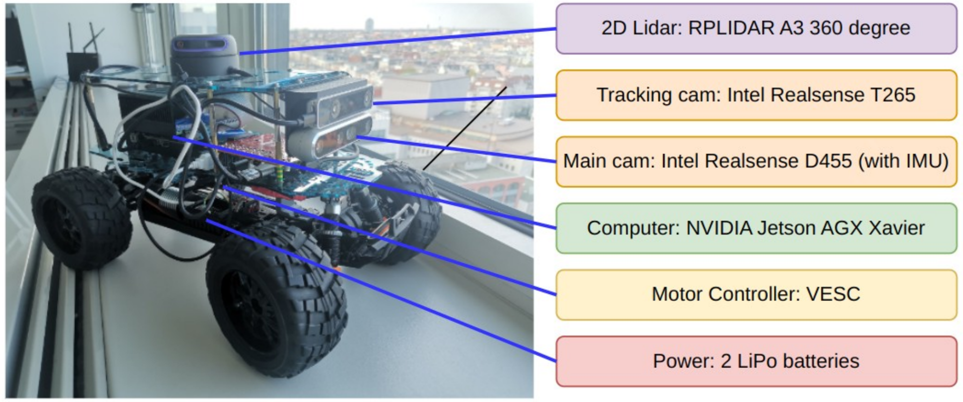
Yuan, Yuchen

Goal

- Implement and deploy Visual SLAM algorithms on Bearcar.
- Evaluate your work in real time scenario.

Prerequisites

- Linux
- ROS2 (Robot Operating System)
- C++ / Python
- NVIDIA GPU (Nice to have)
- Docker (optional)



References

- Isaac ROS Visual SLAM (suggested as the starting point of this project)
- ROS2 (Robot Operating System)
- Jetson AGX Xavier (Main computer of Bearcar)
- rtabmap, rtabmap ROS docker image

AprilTag-based localization

Yuchen Liu, Philipp Grosenick

Introduction

- AprilTag is a type of fiducial marker system used for reliable visual localization in robotics.
- It provides unique IDs and precise pose estimates under varying lighting and angles.
- Ideal for indoor localization and autonomous docking.
- Commonly used in SLAM, calibration, and autonomous navigation tasks.

Resources we provide

- Data recorded from our delivery robot equipped with:
  - 3 Intel RealSense D435 RGBD cameras
  - 2 USB cameras for side view
  - ROS2 Humble
- A Gazebo-based simulation environment for testing and debugging without hardware.

AprilTag-based localization

Yuchen Liu, Philipp Grosenick

Proposed Project Goal

- Build a real-time AprilTag-based localization module. Explore and compare differnet implementation approaches, e.g. OpenCV, deep learning, etc.
- Optionally, experiment with alternative fiducial marker systems, such as ArUco markers, and evaluate their performance.
- Fuse camera-based tag detections with robot odometry for pose estimation.
- Validate localization results with 3d-lidar-based localization.

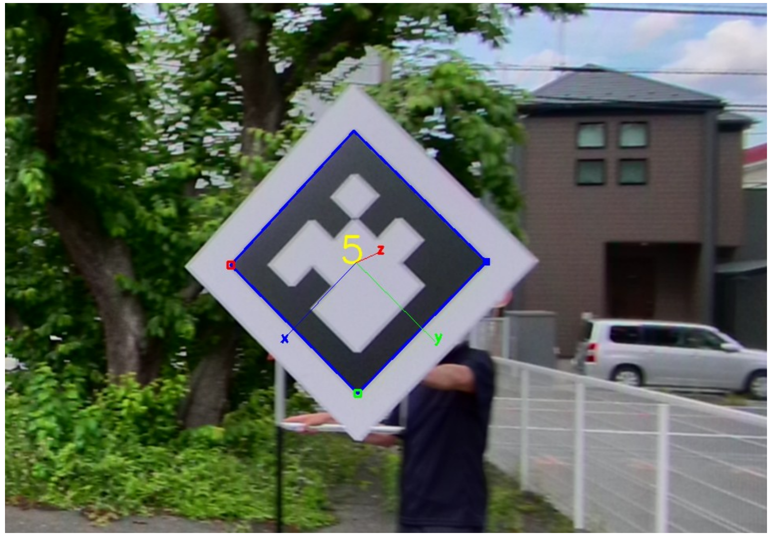


Image source: https://autowarefoundation.github.io/autoware_universe/main/localization/autoware_landmark_based_localizer/autoware_ar_tag_based_localizer/index.html

Prerequisites / Interests

- Linux
- ROS2 (Robot Operating System)
- C++/Python
- Robotics

References

[ROS2 Humble]

[April Tag based localization Autoware]

Driving with Language

Yuan

Incorporating the language modality, this task connects Vision Language Models (VLMs) and autonomous driving systems. The model will introduce the reasoning ability of LLMs to make decisions, and pursue generalizable and explainable driving behavior. Given multi-view images as inputs, models are required to answer questions covering various aspects of driving.

Prerequisites

- Knowledge in machine learning (LLM)
- Python programming skills, deep learning frameworks (Pytorch, Tensorflow)

Reference

- *CVPR 2024 Autonomous Grand Challenge*
- *DriveLM: Driving with Graph Visual Question Answering*
- *LingoQA: Visual Question Answering for Autonomous Driving*

How to evaluate VQA in driving thoroughly

Q: What are the **important** objects in the current scene?

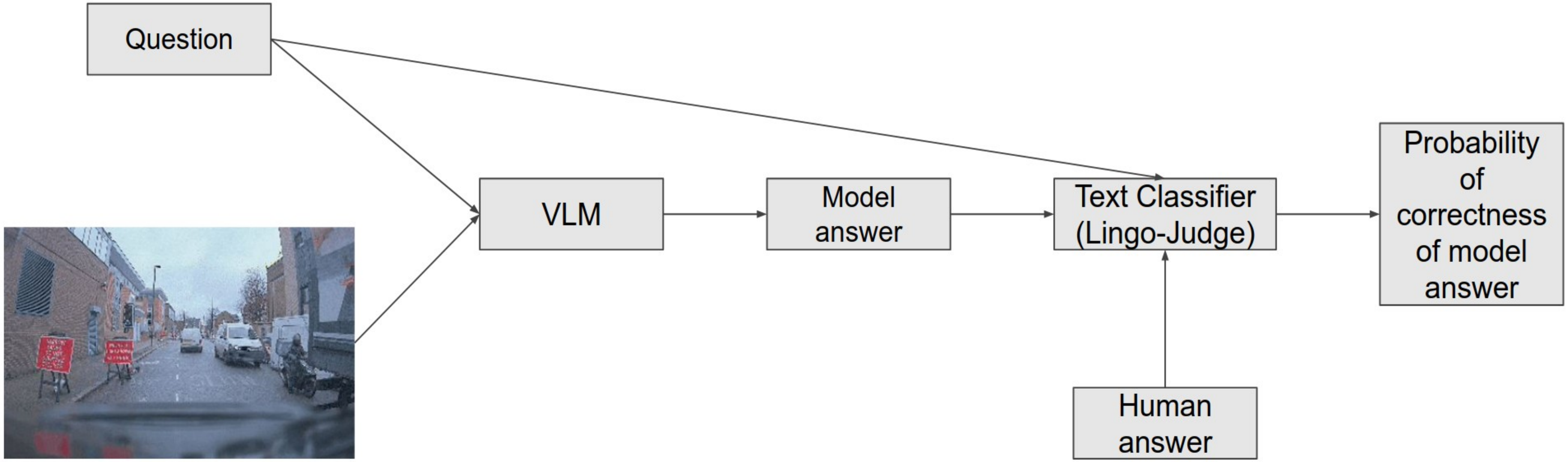
A: There is a brown SUV to the back of the ego vehicle, a black sedan to the back of the ego vehicle, and a green light to the front of the ego vehicle.

Predict: There is a brown SUV to the back of the ego vehicle.

How to evaluate it and reflect its influence to the following QA?

We want to evaluate the correct part “brown SUV”, and penalise the missing parts “black sedan, green light”, and reflect the effect of missing in the following QA (prediction & planning).

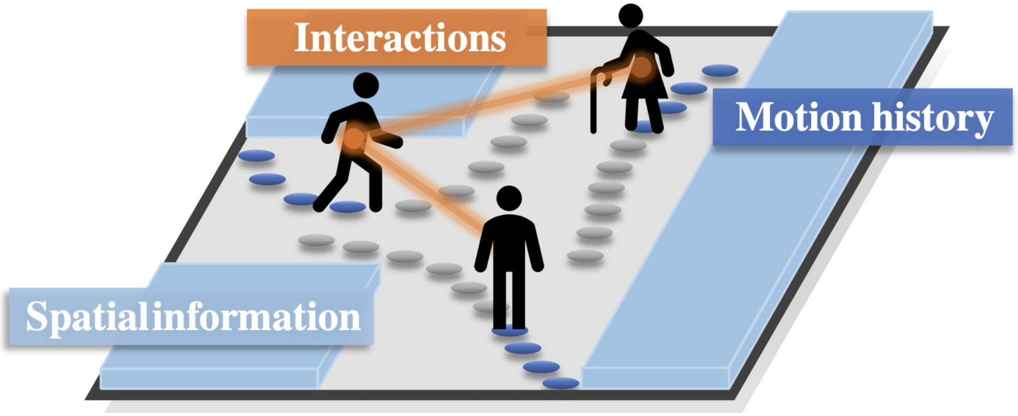
LingoQA Benchmark consists of evaluation metric & dataset



Pedestrian Trajectory Prediction

Evgeny Gorelik

Prediction of pedestrian motion in complex traffic scenarios is an inherently difficult task. An accurate prediction of a persons motion and planned trajectory is necessary in order to plan the ego vehicles path. Improving the quality of trajectories for pedestrians is crucial for better planning capabilities of autonomous vehicles.



Pedestrian Trajectory Prediction

Evgeny Gorelik

The goal of this project is to train a model for prediction of trajectories for pedestrians and vehicles on the road from bounding boxes and/or sensory data alongside map information. Datasets used for training the models can be synthetic or real recordings.

Prerequisites

- Knowledge in machine learning and computer vision
- Python programming skills, deep learning frameworks (Pytorch, Tensorflow)

Reference

- *NuScenes/Waymo/CARLA*
- *[1] Evaluating Pedestrian Trajectory Prediction Methods with Respect to Autonomous Driving*
- *[2] https://github.com/vita-epfl/bounding-box-prediction*