Introduction

- A monocular is a compact refracting telescope used to magnify images of distant objects, typically using an optical prism to ensure an erect image, instead of using relay lenses like most telescopic sights.
- Monocular are ideally suited to those application where three-dimensional perception is not needed, or where compactness and low weight are important.
- The proposed algorithm is motivated by the fact that humans can reliably estimate the scene structure without using binocular vision (with one eye only) by relying on motion parallax and on their vestibular system.
- In this case proprioception is supported by the brain, which uses information from the vestibular system in the head and motion sensing throughout the body to understand the body's translational and rotational kinematics. A similar approach inspired by nature can be applied to machine vision based on monocular camera and motion sensors.
- The simulation results for estimating distance from the camera to a single point are presented below. The purpose of this simulation study is to understand how the estimation accuracy is affected by the following parameters: linear and angular velocity measurement errors, camera noise, and mutual observer and feature point geometry.
- Automated Driving System Toolbox provides a suite of computer vision algorithms that use data from cameras to detect and track objects of interest such as lane markers, vehicles, and pedestrians. Algorithms in the system toolbox are tailored to ADAS and autonomous driving applications
- Object detection is used to locate objects of interest such as pedestrians and vehicles to help perception systems automate braking and steering tasks. The system toolbox provides functionality to detect vehicles, pedestrians, and lane markers through pretrained detectors using machine learning, including deep learning, as well as functionality to train custom detectors

Working

Define Camera Configuration:

Knowing the camera’s intrinsic and extrinsic calibration parameters is critical to accurate conversion between pixel and vehicle coordinates. Start by defining the camera’s intrinsic parameters. The parameters below were determined earlier using a camera calibration procedure that used a checkerboard calibration pattern. You can use the camera Calibrator app to obtain them for your camera.

Load a Frame of Video:

Before processing the entire video, process a single video frame to illustrate the concepts involved in the design of a monocular camera sensor. Start by creating a VideoReader object that opens a video file. To be memory efficient, VideoReader loads one video frame at a time.

Create Bird's-Eye-View Image:

There are many ways to segment and detect lane markers. One approach involves the use of a bird's-eye-view image transform. Although it incurs computational cost, this transform offers one major advantage. The lane markers in the bird's-eye view are of uniform thickness, thus simplifying the segmentation process. The lane markers belonging to the same lane also become parallel, thus making further analysis easier.

Find Lane Markers in Vehicle Coordinates:

Having the bird's-eye-view image, you can now use the segmentLaneMarkerRidge function to separate lane marker candidate pixels from the road surface. This technique was chosen for its simplicity and relative effectiveness. Alternative segmentation techniques exist including semantic segmentation (deep learning) and steerable filters. We substitute these techniques below to obtain a binary mask needed for the next stage.

Determine Boundaries of the Ego Lane:

Remove additional boundaries based on the strength metric computed by the findParabolicLaneBoundaries function. Set a lane strength threshold based on ROI and image size.

Locate Vehicles in Vehicle Coordinates:

Use the configureDetectorMonoCamera function to specialize the generic ACF detector to take into account the geometry of the typical automotive application. By passing in this camera configuration, this new detector searches only for vehicles along the road's surface, because there is no point searching for vehicles high above the vanishing point. This saves computational time and reduces the number of false positives.

Applications: