Hardware for Self-driving Cars

A. Giavaras

Contents

1 Hardware for Self-driving Cars

In this chapter, we will discuss sensors, and the various types of them available for the task of perception. Next we will discuss the self-driving car hardware available nowadays.

1.1 Sensors

Let's begin by talking about sensors. Even the best perception algorithms are limited by the quality of their sensor data. And careful selection of sensors can go a long way to simplifying the self-driving perception task. Let's try to give a definition of what a sensor is.

Definition 1.1. What is a sensor?

For our purposes, a sensor is any device that measures or detects some property of the environment, or changes to that property over time.

Sensors are broadly categorized into two types, depending on what property they record. If they record a property of the environment they are called **exteroceptive**. Extero means outside, or from the surroundings. On the other hand, if the sensors record a property of the ego vehicle, they are called **proprioceptive**. Proprios means internal, or one's own. Let's start by discussing common exteroceptive sensors.

1.1.1 Camera

We start with the most common and widely used sensor in autonomous driving, the camera. Cameras are a passive, light-collecting sensor that are great at capturing rich, detailed information about a scene. In fact, some groups believe that the camera is the only sensor truly required for self-driving. But state of the art performance is not yet possible with vision alone. While talking about cameras, we usually tend to talk about three important comparison metrics. We select cameras in terms:

- resolution
- field of view or FOV
- dynamic range

The resolution is the number of pixels that create the image. So it's a way of specifying the quality of the image. The field of view is defined by the horizontal and vertical angular extent that is visible to the camera, and can be varied through lens selection and zoom. The dynamic range of the camera is the difference between the darkest and the lightest tones in an image. High dynamic range is critical for self-driving vehicles due to the highly variable lighting conditions encountered while driving especially at night.

There is an important trade off cameras and lens selection, that lies between the choice of field of view and resolution. Wider FOV permits a lager viewing region in the environment, but fewer pixels that absorb light from one particular object. As the FOV increases, we need to increase resolution to still be able to perceive with the same quality, the various kinds of information we may encounter. Other properties of cameras that affect perception exist as well, such as focal length, depth of field and frame rate.

The combination of two cameras with overlapping fields of view and aligned image planes is called the stereo camera. Stereo cameras allow depth estimation from synchronized image pairs. Pixel values from image can be matched to the other image producing a disparity map of the scene. This disparity can then be used to estimate depth at each pixel.

1.1.2 LIDAR

Next we have LIDAR which stands for light detection and ranging sensor. LIDAR sensing involves shooting light beams into the environment and measuring the reflected return. By measuring the amount of returned light and time of flight of the beam. Both in intensity in range to the reflecting object can be estimated. An illustration of LIDAR based environment representation is shown in figure 1.

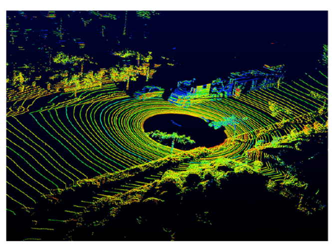

Fig. 1: LIDAR illustration of environment representation.

LIDAR usually includes a spinning element with multiple stacked light sources and outputs a three dimensional point cloud map, which is great for assessing scene geometry. Because it is an active sensor with it's own light sources, LIDAR are not effected by the environments lighting. So LIDAR do not face the same challenges as cameras when operating in poor or variable lighting conditions. Let's discuss the important comparison metrics for selecting LIDAR.

- The first is the number of sources it contains with 8, 16, 32, and 64 being common sizes.

- the second is the points per second it can collect. The faster the point collection, the more detailed the 3D point cloud can be.

Another characteristic is the rotation rate. The higher this rate, the faster the 3D point clouds are updated. Detection range is also important, and is dictated by the power output of the light source. And finally, we have the field of view, which once again, is the angular extent visible to the LIDAR sensor.

Finally, we should also mention the new LIDAR types that are currently emerging. High-resolution, solid-state LIDAR. Without a rotational component of the typical LIDARs, these sensors stand to become extremely low-cost and reliable. Thanks to being implemented entirely in silicon. HD solid-state LIDAR are still a work in progress. But definitely something exciting for the future of affordable self-driving.

1.1.3 Radar

Our next sensor is RADAR, which stands for radio detection and ranging. RADAR sensors have been around longer than LIDAR and robustly detect large objects in the environment. They are particularly useful in adverse weather as they are mostly unaffected by precipitation. Let's discuss some of the comparison metrics for selecting RADAR. RADAR are selected based on

- detection range
- field of view,
- the position and speed measurement accuracy.

RADARs are also typically available as either having a wide angular field of view but short range. Or having a narrow FOV but a longer range.

1.1.4 Ultrasonics or sonars

The next sensor we are going to discuss are ultrasonics or sonars. Originally so named for sound navigation and ranging. Which measure range using sound waves. Sonars are sensors that are short range and inexpensive ranging devices. This makes them good for parking scenarios, where the ego-vehicle needs to make movements very close to other cars. Another great thing about sonar is that they are low-cost. Moreover, just like RADAR and LIDAR, they are unaffected by lighting and precipitation conditions. A sonar sensor is selected based on a few key metrics itemized next.

- The maximum range they can measure
- The the detection FOV
- The cost

1.1.5 Proprioceptive sensors

Now let's discuss the proprioceptive sensors, the sensors that sense ego properties. The most common ones here are:

- Global Navigation Satellite Systems, GNSS for short, such as GPS or Galileo
- Inertial Measurement Units or IMU's
- Wheel odometers

GNSS receivers are used to measure ego vehicle position, velocity, and sometimes heading. The accuracy depends a lot on the actual positioning methods and the corrections used. Apart from these, the IMU also measures the angular rotation rate, accelerations of the ego vehicle, and the combined measurements can be used to estimate the 3D orientation of the vehicle. Where heading is the most important for vehicle control. Finally, we have wheel odometry sensors. This sensor tracks the wheel rates of rotation, and uses these to estimate the speed and heading rate of change of the ego car. This is the same sensor that tracks the mileage on your vehicle.

In summary, the major sensors used nowadays for autonomous driving perception include cameras, RADAR, LIDAR, sonar, GNSS, IMUs, and wheel odometry modules. These sensors have many characteristics that can vary wildly, including resolution, detection range, and FOV.

Selecting an appropriate sensor configuration for a self-driving car is not trivial. Figure 2 is a simple graphic that shows each of the sensors and where they usually go on a car.

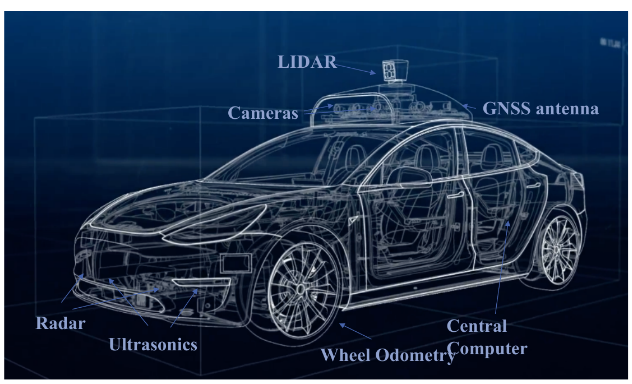

Fig. 2: Usual sensor positioning.

1.2 Computing Hardware

Let's now discuss a little bit about the computing hardware most commonly used in self-driving cars at the time of writing. The most crucial part is the computing brain, the main decision making unit of the car. It takes in all sensor data and outputs the commands needed to drive the vehicle. Most companies prefer to design their own computing systems that match the specific requirements of their sensors and algorithms. Some hardware options exist, however, that can handle self-driving computing loads out of the box.

The most common examples would be Nvidia's Drive PX and Intel & Mobileye's EyeQ. Any computing brain for self-driving needs both serial and parallel compute modules. Particularly for image and LIDAR processing to do segmentation, object detection, and mapping. For these we employ GPUs, FPGAs and custom ASICs (Application Specific Integrated Circuit), which are specialized hardware to do a specific type of computation.

For example, the drive PX units include multiple GPUs. The EyeQs have FPGAs both to accelerate parallalizable compute tasks, such as image processing or neural network inference.

Finally, a quick comment about synchronization. Because we want to make driving decisions based on a coherent picture of the road scene. It is essential to correctly synchronize the different modules in the system, and serve a common clock. Fortunately, GPS relies on extremely accurate timing to function, and as such can act as an appropriate reference clock when available. Regardless, sensor measurements must be timestamped with consistent times for sensor fusion to function correctly. Let's summarize. In this video, we learned about sensors and their different types based on what they measure.

1.3 Hardware Configuration

Section 1.1, covers the various kinds of sensors most commonly used for perception. One question that should be answered is how do we place these sensors on the vehicle in order to aquire a complete view of the environment?

In this section, we will discuss the configuration design to meet sensor coverage needs for an autonomous driving car. We will do this by going through two common scenarios:

- Driving on a highway and
- Driving in an urban environment

After analyzing these scenarios, we will lay out the overall coverage requirements and discuss some issues with the design.

Let's however begin by recalling the most commonly available sensors. These are:

- The camera for appearance input.
- The stereo camera for depth information
- Lidar for all whether 3D input
- Radar for object detection
- Ultrasonic for short-range 3D input
- GNSS/IMU data and wheel odometry for ego state estimation.

Also, remember that all of these sensors come in different configurations and different ranges in FOV over which they can sense. They have some resolution that depends on the instrument specifics and the field of view. Before we move to discussing coverage, let's define the deceleration rates we're willing to accept for driving which will drive the detection ranges needed for our sensors.

Remark 1.1. **Aggressive Deceleration**

Aggressive deceleration is set to $5m/sec^2$ which is roughly the deceleration you experience when you slam the brakes hard and try to stop abruptly in case of an emergency.

Normal decelerations are set to $2m/sec^2$, which is reasonably comfortable while still allowing the car to come to a stop quickly. Given a constant deceleration our braking distance d can be computed as follows according to equation 1.

$$d = \frac{V^2}{2\alpha} \tag{1}$$

where V is the vehicle velocity and α is its rate of deceleration. We can also factor in reaction time of the system and road surface friction limits, but we'll keep things simple in this discussion.

Let's talk about coverage now. The question we want to answer is where should we place our sensors so that we have sufficient input for our driving task? Practically speaking, we want our sensors to capture the ODD we have in mind or the ODD our system can produce decisions for. We should be able to provide all of the decisions with sufficient input. There can be so many possible scenarios in driving but we'll look at just two common scenarios to see how the requirements drive our sensor selection. Will look at highway and urban driving. Let's think about these two situations briefly.

For a divided highway, we have fast moving traffic, usually high volume, and quite a few lanes to monitor, but all vehicles are moving in the same direction. The other highlight of driving on a highway setting is that there are fewer and gradual curves and we have exits and merges to consider as well.

On the other hand, in the urban situation we'll consider, we have moderate volume and moderate speed traffic with fewer lanes but with traffic moving in all directions especially through intersections.

1.3.1 Highway Scenario

Let's start with the highway setting. We can break down the highway setting into three basic maneuver needs.

- We may need to hit the brakes hard if there's an emergency situation.
- We need to maintain a steady speed matching the flow of traffic around us.
- We might need to change lanes.

In the case of an emergency stop, if there is a blockage on our road we want to stop in time. So, applying our stopping distance equation longitudinally, we need to be able to sense about a 110 meters in front of us assuming a highway speed of a 120 kilometers and aggressive deceleration. Most self-driving systems aim for sensing ranges of a 150 to 200 meters in front of the vehicle as a result. Similarly, to avoid lateral collision or to change lanes to avoid hitting an obstacle in our lane, we need to be able to sense at least our adjacent lanes, which are 3.7 meters wide in North America. To maintain speed during vehicle following, we need to sense the vehicle in our own lane. Both their relative position and the speed are important to maintain a safe following distance. This is usually defined in units of time for human drivers and set to two seconds in nominal conditions. It can also be assessed using aggressive deceleration of the lead vehicle and the reaction time from our ego vehicle. So, at a 120 kilometers per hour, relative position and speed measurements to a range of 165 meters are needed and typical systems use 100 meters for this requirement. Laterally, we need to know what's happening anywhere in our adjacent lanes in case another vehicles seeks to merge into our lane or we need to merge with other traffic. A wide 160 to 180 degree field of view is required to track adjacent lanes and a range of 40 to 60 meters is needed to find space between vehicles.

Finally, let's discuss the lane change maneuver and consider the following scenario. Suppose we want to move to the adjacent lane, longitudinally we need to look forward, so we are a safe distance from the leading vehicle and we also need to look behind just to see what the rear vehicles are doing and laterally it's a bit more complicated. We may need to look beyond just the adjacent lanes. For example, what if a vehicle attempts to maneuver into the adjacent lane at the same time as we do? We'll need to coordinate our lane change room maneuvers so we don't crash.

The sensor requirements for lane changes are roughly equivalent to those in the maintain speed scenario. As both need to manage vehicles in front of and behind the ego vehicle as well as to each side. Overall, this gives us the picture for coverage requirements for the highway driving scenario. We need longitudinal sensors and lateral sensors and both wide and narrow FOV sensors to do these three maneuvers, the emergency stop, maintaining speed and changing lanes. Already from this small set of ODD requirements we see a large variety of sensor requirements that arise.

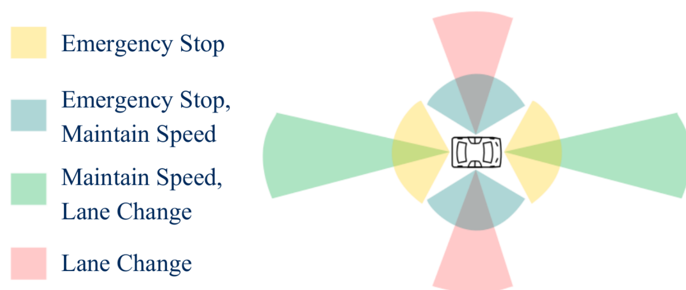

Fig. 3: Highway analysis overall coverage.

1.3.2 Urban Scenario

Let's discuss the urban scenario next. The urban scenario as we discussed before is a moderate volume, moderate traffic scenario with fewer lanes on the highway case but with the added complexity of pedestrians. There are six types of basic maneuvers here. Obviously, we can still perform emergency stop, maintain speed and lane changes but we also have scenarios such as overtaking a parked car, left and right turns at intersections and more complex maneuvers through intersections such as roundabouts. In fact, for the first three basic maneuvers, the coverage analysis is pretty much the same as the highway analysis but since we are not moving as quickly, we don't need the same extent for our long-range sensing.

Let's discuss the overtake maneuver next. More specifically, consider a case where you have to overtake a parked car. Longitudinally, we definitely need to sense the parked car as well as look for oncoming traffic. So, we need both sensors, wide short-range sensors to detect the parked car and narrow long-range sensors to identify if oncoming traffic is approaching. Laterally, we'll need to observe beyond the adjacent lanes for merging vehicles as we did in the highway case. Intersections require that we have near omni-directional sensing for all kinds of movements that can occur. Approaching vehicles, nearby pedestrians, doing turns and much more. Finally, for roundabouts we need a wide-range, short distance sensor laterally since the traffic is slow but we also need a wide-range short distance sensor longitudinally because of how movement around the roundabout occurs. We need to sense all of the incoming traffic flowing through the roundabout to make proper decisions.

Thus, we end up with the overall coverage diagram for the urban case shown in figure 4. The main difference with respect to highway coverage is because of the sensing we require for movement at intersections and at roundabouts and for the overtaking maneuver. In fact, the highway case is almost entirely covered by the urban requirements.

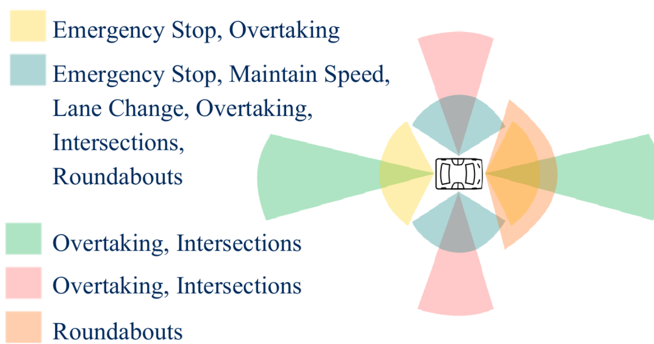

Fig. 4: Urban analysis overall coverage.

Let's summarize the coverage analysis. For all of the maneuvers we do, we need long range sensors which typically have shorter angular field of view and wide angular field of view sensors which typically have medium to short-range sensing. As the scenarios become more complex, we saw the need for full 360 degrees sensor coverage on the short scale out to about 50 meters and much longer range requirements in the longitudinal direction. We can also add even shorter range sensors like sonar which are useful in parking scenarios and so in the end our sensor configuration looks something like this diagram.

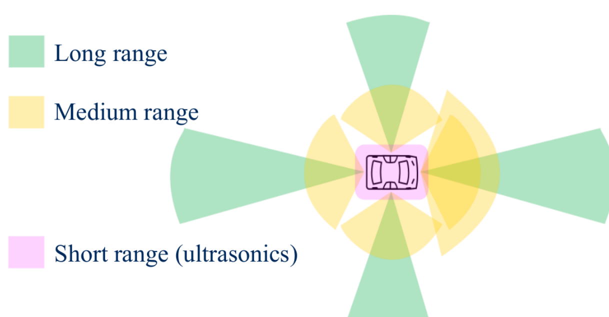

Fig. 5: Overall coverage.

To summarize, our choice of sensors should be driven by the requirements of the maneuvers we want to execute and it should include both long-range sensors for longitudinal dangers and wide field of view sensors for omnidirectional perception. The final choice of configurations also depends on our requirements for operating conditions, sensor redundancy due to failures and on budget. There is no single answer to which sensors are needed for a self-driving car.

2 Questions

1. What are the differences between exteroceptive sensors and proprioceptive sensors? (Select all that apply)
   (a) Proprioceptive sensors do not interact with the environment, whereas exteroceptive sensors do.
   (b) Proprioceptive sensors can determine distance traveled by the vehicle, whereas exteroceptive sensors cannot.
   (c) Exteroceptive sensors can determine obstacle size and distance away, whereas proprioceptive sensors cannot.
   (d) Proprioceptive sensors are used to determine vehicle position, whereas exteroceptive sensors are used for sensing the environment.
   (e) Exteroceptive sensors can determine distance traveled by the vehicle, whereas proprioceptive sensors cannot.
2. Which of the following exteroceptive sensors would you use in harsh sunlight?
   (a) Sonar
   (b) Cameras
   (c) Radar
   (d) Lidar
3. Why is synchronization and timing accuracy important in the self driving system? Choose the primary reason.
   (a) Synchronization is important to ensure that sensors measure the environment at the same time.
   (b) Synchronization is important to ensure organized computation.
   (c) Synchronization is important to ensure correct sensor fusion.
   (d) Synchronization is important to check sensor failure.
4. Your autonomous vehicle is driving on the German autobahn at $150 km/h$ and you wish to maintain safe following distances with other vehicles.

Assuming a safe following distance of $2s$, what is the distance (in m) required between vehicles? Round your answer to 2 decimal places.

5. Using the same speed of $150km/h$, what is the braking distance (in m) required for emergency stops? Assume an aggressive deceleration of $5m/s^2$. Round your answer to two decimal places.

6. Suppose your vehicle was using long range cameras for sensing forward distance, but it is now nighttime and the images captured are too dark. Which of the following sensors can be used to compensate?

   (a) Lidar

   (b) IMU

   (c) Radar

   (d) Sonar

7. What are the differences between an occupancy grid and a localization map? (Select all that apply)

   (a) The localization map is primarily used to estimate the vehicle position, whereas the occupancy grid is primarily used to plan collision free paths.

   (b) The localization map uses only lidar data, whereas the occupancy grid can use both lidar and camera data.

   (c) An occupancy grid uses a dense representation of the environment, whereas a localization map does not need to be dense.

   (d) The occupancy grid only contains static objects, while the localization map contains only dynamic objects.

8. The vehicle steps through the software architecture and arrives at the controller stage. What information is required for the controller to output its commands to the vehicle?

   (a) Planned paths

   (b) Vehicle state

   (c) Locations of obstacles and other vehicles

   (d) Environment maps

9. What is (are) the role(s) of the system supervisor? (Select all that apply)

   (a) To ensure that the maps update at the correct frequencies

   (b) To ensure that the planned paths are collision free

   (c) To ensure that the controller outputs are within operating range

   (d) To ensure that the sensors are working correctly

10. Which of the following tasks should be assigned to the local planner?
    (a) Planning a merge onto the highway
    (b) Planning a route to a destination
    (c) Planning a lane change to turn left
    (d) Planning to avoid a parked car in the ego vehicle's lane
11. What common objects in the environment appear in the occupancy grid?
    (a) Other moving vehicles
    (b) Lane boundaries
    (c) Parked vehicles
    (d) Traffic lights
12. Which of the following maps contain roadway speed limits?
    (a) Occupancy grid
    (b) Localization map
    (c) Detailed roadmap

3 Sensor Calibration

Now that we've seen how we can combine multiple sources of sensor data to estimate the vehicle state, it's time to address the topic of **sensor calibration**. Sensor calibration and is absolutely essential for doing state estimation properly. Concretely, in this section, we will discuss the three main types of sensor calibration and why we need to think about them when designing a state estimator for a self-driving car. The three main types of calibration will talk about are

- Intrinsic calibration, which deals with sensors specific parameters
- Extrinsic calibration, which deals with how the sensors are positioned and oriented on the vehicle
- Temporal calibration, which deals with the time offset between different sensor measurements.

3.1 Intrinsic calibration

Let's look at intrinsic calibration first. In intrinsic calibration, we want to determine the fixed parameters of our sensor models, so that we can use them in an estimator like an extended Kalman filter. Every sensor has parameters associated with it that are unique to that specific sensor and are typically expected to be constant.

For example, we might have an encoder attached to one axle of the car that measures the wheel rotation rate ω. If we want to use ω to estimate the forward velocity v of the wheel, we would need to know the radius R of the wheel, so that we can use the following equation

$$v = \omega R \tag{2}$$

In this case, R is a parameter of the sensor model that is specific to the wheel the encoder is attached to and we might have a different R for a different wheel. Another example of an intrinsic sensor parameter is the elevation angle of a scan line in a LiDAR sensor like the Velodyne. The elevation angle is a fixed quantity but we need to know it ahead of time so that we can properly interpret each scan.

So, how do we determine intrinsic parameters like these? Well, there are a few practical strategies for doing this. The easiest one is just let the manufacturer do it for you. Often, sensors are calibrated in the factory and come with a spec sheet that tells you all the numbers you need to plug into your model to make sense of the measurements. This is usually a good starting point but it will not always be good enough to do really accurate state estimation because no two sensors are exactly alike and there will be some variation in the true values of the parameters. Another easy strategy that involves a little more work is to try measuring these parameters by hand. This is pretty straightforward for something like a tire, but not so straightforward for something like a LiDAR where it's not exactly practical to poke around with a protractor inside the sensor.

A more sophisticated approach involves estimating the intrinsic parameters as part of the vehicle state, either on the fly or more commonly as a special calibration step before putting the sensors into operation. This approach has the advantage of producing an accurate calibration that's specific to the particular sensor and can also be formulated in a way that can handle the parameters varying slowly over time.

For example, if you continually estimate the radius of your tires, this could be a good way of detecting when you have a flat. Now, because the estimators we've talked about in this course are general purpose, we already have the tools to do this kind of automatic calibration. In order to see how this works, let's come back to our example of a car moving in one dimension.

we have attached an encoder to the back wheel to measure the wheel rotation rate. If we want to estimate the wheel radius along with position and velocity, all we need to do is add it to the state vector and work out what the new motion and observation model should be. For the motion model, everything is the same as before except now there's an extra row and column in the matrix that says that the wheel radius should stay constant from one time step to the next. For the observation model, we are still observing position directly through GPS but now we're also observing the wheel rotation rate through the encoder. So, we include the extra non-linear observation in the model. From here, we can use the extended or unscented Kalman filter to estimate the wheel radius along with the position and velocity of the vehicle.

Thus, intrinsic calibration is essential for doing state estimation with even a single sensor.

3.2 Extrinsic calibration

Extrinsic calibration is equally important for fusing information from multiple sensors. In extrinsic calibration, we are interested in determining the relative poses of all of the sensors usually with respect to the vehicle frame.

For example, we need to know the relative pose of the IMU and the LiDAR. The rates reported by the IMU are expressed in the same coordinate system as the LiDAR point clouds. Just like with intrinsic calibration, there are different techniques for doing extrinsic calibration. If you are lucky, you might have access to an accurate CAD model of the vehicle, where all of the sensor frames have been nicely laid out for you. If you are less lucky, you might be tempted to try measuring by hand. Unfortunately, this is often difficult or impossible to do accurately since many sensors have the origin of their coordinate system inside the sensor itself, and you probably do not want to dismantle your car and all of the sensors.

Fortunately, we can use a similar trick to estimate the extrinsic parameters by including them in our state. This can become a bit complicated for arbitrary sensor configurations, and there is still a lot of research being done into different techniques for doing this reliably.

3.3 Temporal calibration

Finally, an often overlooked but still important type of calibration is temporal calibration. In all of our discussion of multisensory fusion, we've been implicitly assuming that all of the measurements we have combined are captured exactly the same moment in time or at least close enough for a given level of accuracy. But how do we decide whether two measurements are close enough to be considered synchronized? Well, the obvious thing to do would just be to timestamp each measurement when the on-board computer receives it, and match up the measurements that are closest to each other.

For example, if we get LiDAR scans at 15 hertz and IMU readings at 200 hertz, we might want to pair each LiDAR scan with the IMU reading whose timestamp is the closest match.

In reality, there is an unknown delay between when the LiDAR or IMU actually records an observation and when it arrives at the computer. These delays can be caused by the time it takes for the sensor data to be transmitted to the host computer, or by pre-processing steps performed by the sensor circuitry, and the delay can be different for different sensors. Hence, if we want to get a really accurate state estimate, we need to think about how well our sensors are actually synchronized, and there are different ways to approach this. The simplest and most common thing to do is just to assume the delay is zero. You can still get a working estimator this way, but the results may be less accurate than what you would get with a better temporal calibration.

Another common strategy is to use hardware timing signals to synchronize the sensors, but this is often an option only for more expensive sensor setups. As you may have guessed, it's also possible to try estimating these time delays as part of the vehicle state, but this can get complicated.

To summarize, sensor fusion is impossible without calibration. In this section, we touched upon three types of calibration. Intrinsic calibration, which deals with calibrating the parameters of our sensor models. Extrinsic calibration, which gives us the coordinate transformations we need to transform sensor measurements into a common reference frame. Temporal calibration, which deals with synchronizing measurements to ensure they all correspond to the same vehicle state. While there are some standard techniques for solving all of these problems, calibration is still very much an active area of research.

4 Questions

1. What is intrinsic calibration?
2. Why do we need extrinsic calibration?
3. What is temporal calibration?

5 Answers to Questions

Answer to questions in section 2

1. What are the differences between exteroceptive sensors and proprioceptive sensors? (Select all that apply)
2. Which of the following exteroceptive sensors would you use in harsh sunlight?
3. Why is synchronization and timing accuracy important in the self driving system? Choose the primary reason.
4. Your autonomous vehicle is driving on the German autobahn at $150km/h$ and you wish to maintain safe following distances with other vehicles. Assuming a safe following distance of $2s$, what is the distance (in m) required between vehicles? Round your answer to 2 decimal places.
5. Using the same speed of $150km/h$, what is the braking distance (in m) required for emergency stops? Assume an aggressive deceleration of $5m/s^2$. Round your answer to two decimal places.
6. Suppose your vehicle was using long range cameras for sensing forward distance, but it is now nighttime and the images captured are too dark. Which of the following sensors can be used to compensate?
7. What are the differences between an occupancy grid and a localization map? (Select all that apply)
8. The vehicle steps through the software architecture and arrives at the controller stage. What information is required for the controller to output its commands to the vehicle?
9. What is (are) the role(s) of the system supervisor? (Select all that apply)
10. Which of the following tasks should be assigned to the local planner?
11. What common objects in the environment appear in the occupancy grid?
12. Which of the following maps contain roadway speed limits?

References

[1] SAE *Taxonomy of Driving* $https://www.sae.org/standards/content/j3016\_201806/?PC = DL2BUY$

[2] SAE *SAE J3016 Taxonomy and Definitions Document* $https://drive.google.com/open?id = 1xtOqFVJvOElXjXqf4RAwXZkI\_EwbxFMg$

[3] Åström K. J., Murray R. M. *Feedback Systems. An Introduction for Scientists and Engineers*

[4] Philip , Florent Altche1, Brigitte dAndrea-Novel, and Arnaud de La Fortelle *The Kinematic Bicycle Model: a Consistent Model for Planning Feasible Trajectories for Autonomous Vehicles?* HAL Id: hal-01520869, https://hal-polytechnique.archives-ouvertes.fr/hal-01520869

[5] Marcos R. O., A. Maximo *Model Predictive Controller for Trajectory Tracking by Differential Drive Robot with Actuation constraints*