18
Sensor Fusion

Almost every robot will rely on multiple sensors (including multiple *types* of sensors) for perception and localization tasks. This allows the robot to take advantage of the different strengths of each sensor for a more well-rounded sensing capability. For example a self-driving car may use both laser rangefinders and radar for measuring distances, since in some cases one sensor may work better than the other. As another example, a wheeled robot may use GNSS sensors as well as wheel encoders to estimate position. However, while each sensor may provide data toward a similar goal (e.g. estimating position or orientation) their sensing modalities may be drastically different. This chapter covers the topic of *sensor fusion*[1,2], and provides a discussion on algorithms for effectively leveraging multiple sensing modalities toward a common objective.

[1] F. Gustafsson. *Statistical Sensor Fusion*. Studentlitteratur, 2013, p. 554

[2] D. Simon. *Optimal State Estimation: Kalman, H_∞, and Nonlinear Approaches*. John Wiley & Sons, 2006

Sensor Fusion

Using measurements from multiple sensors (potentially different types of sensors) is an effective technique for reducing the uncertainty in downstream perception and estimation tasks (see Figure 18.1). This is generally the case because

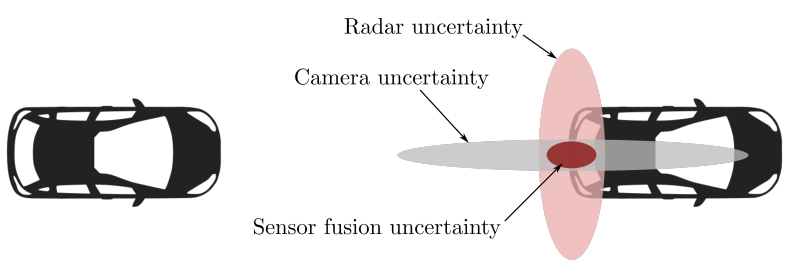

Figure 18.1: Sensor fusion can reduce uncertainty by providing more well-rounded data. For example in this scenario the radar sensor may have good accuracy longitudinally but less accuracy laterally. Contrarily, a camera may provide poor range estimation but good lateral position estimation. By fusing these two sensor measurements the resulting estimate can be accurate both longitudinally and laterally.

individual sensors typically suffer from limited range, limited field of view, or performance degradation under certain environmental conditions. Additionally, in single-sensor systems measurement accuracy degradation and sensor failure can be catastrophic. Alternatively, multi-sensor systems can address these challenges through redundancy of individual sensors (e.g. to provide full field of view measurements or multiple measurements of the same quantity) or through

sensor diversity (e.g. using sensors with different characteristics to offset limitations of others).

18.1 A Taxonomy of Sensor Fusion

To put the sensor fusion problem into a broader perspective, a taxonomy of sensor fusion related challenges will now be presented. This includes challenges associated with both fusion algorithms as well as the measurement data.

18.1.1 Data-related Taxonomy

One of the primary challenges with data fusion is the inherent imperfection in the measurement data, including uncertainty (i.e. resulting from sensor noise), imprecision (i.e. resulting from sensor bias), and granularity (i.e. resulting from sensor resolution). Other important data-related aspects to sensor fusion include data correlation, disparity, and inconsistency (e.g. data conflicts, outliers, disorder). Broadly speaking sensor data can experience multiple types of imperfection at the same time, and so data fusion algorithms should be developed with robustness in mind.

18.1.2 Fusion-related Taxonomy

At the data-fusion level, it is useful to classify the problem based on the type of data that is being fused. Low-level fusion problems typically fuse low-level signal data (i.e. time-series data), intermediate-level problems fuse features and characteristics, and high-level fusion problems consider decisions. Fusion problems can also be categorized based on the relationship among different sensors used in the fusion process. Competitive fusion problems consider redundant sensors that directly measure the same quantity. Complementary fusion is used when different sensors provide complementary information about the environment (e.g. lidar for short distance ranging and radar for long distance ranging). Finally, cooperative fusion considers problems where the required information cannot be inferred from a single sensor (e.g. GNSS localization and stereo vision can be cooperatively used because they measure fundamentally different environmental quantities). Generally speaking competitive fusion increases reliability and accuracy of fused information, complementary fusion increases the completeness of information, and cooperative fusion broadens the types of information that can be gathered.

18.1.3 Architectural Taxonomy

Fusion algorithms can also be classified based on their type of architecture, namely whether they are centralized, decentralized, or distributed. Centralized architectures collect *all* sensor data first, and then perform computations on the entire set of data. This approach is theoretically optimal since all information is

gathered and operated on at once, but the need for high levels of communication and processing can be challenging in practice. Decentralized architectures are essentially collections of centralized systems, and generally still suffer from the same high demands for communication and processing. On the other hand, distributed architectures do not collect all sensor information ahead of time but rather perform computations on local sensor data first, before potentially passing information on for further fusion tasks. These architectures scale better, but can lead to suboptimal performance because each sensor is performing local processing (i.e. without having all information).

18.2 Bayesian Approach to Sensor Fusion

Previous chapters presented several algorithms for robot state estimation and localization based on Bayes' filter. In fact, these algorithms can be viewed as approaches to solve the sensor fusion problem. This section explores the Bayesian approach to sensor fusion in more detail to show exactly how these approaches can blend measurement data to reduce uncertainty.

Recall that the Bayesian approach is a probabilistic approach that models unknowns as random variables and quantifies knowledge in the form of probability distributions over the unknowns. This principled approach is useful for sensors fusion for several reasons. First, it provides a unified framework for representing knowledge that is compatible with any quantity and type of sensors and is interpretable. Second, probability distributions implicitly provide information about uncertainty (e.g. the variance of a Gaussian). Third, Bayes' rule provides a principled approach for updating distributions. Finally, they can be used to deal with missing information and classification of new observations.

Example 18.2.1 (Competitive Fusion Example)**.** As an example to show how a probabilistic approach can be used to reduce uncertainty through sensor fusion, consider a case where two sensors are fused to estimate a single quantity $x \in \mathbb{R}$. Specifically, suppose the two measurements y_1 and y_2 are normally distributed random variables:

$$p(y_1 \mid x) = \frac{1}{\sqrt{2\pi\sigma_1^2}} e^{-\frac{1}{2}\frac{(x-y_1)^2}{\sigma_1^2}},$$

$$p(y_2 \mid x) = \frac{1}{\sqrt{2\pi\sigma_2^2}} e^{-\frac{1}{2}\frac{(x-y_2)^2}{\sigma_2^2}},$$

where the first sensor has a higher precision than the second sensor such that $\sigma_1^2 < \sigma_2^2$. Then the *combined* measurement probability is given by:

$$p(y_1, y_2 \mid x) = p(y_1 \mid x)p(y_2 \mid x),$$

by assuming conditional independence. By exploiting the product of two Gaus-

sian property this joint probability distribution is:

$$p(y_1, y_2 \mid x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}},$$

where:

$$\mu = \frac{y_1\sigma_2^2 + y_2\sigma_1^2}{\sigma_1^2 + \sigma_2^2}, \quad \sigma = \frac{\sigma_1^2\sigma_2^2}{\sigma_1^2 + \sigma_2^2}.$$

Therefore, given two measurements y_1 and y_2 the best estimate of the quantity x is given by μ, which is a *weighted* average of the two measurements. In particular, more weight is given to the measurement with higher precision (i.e. higher variance σ_i^2) and the overall uncertainty will decrease!

18.2.1 Kalman Filter Sensor Fusion

The Kalman filter from the previous chapter on parametric state estimation techniques is a common tool for sensor fusion problems. Recall that the Kalman filter assumes a linear state transition (dynamics) model:

$$\boldsymbol{x}_t = A_t \boldsymbol{x}_{t-1} + B_t \boldsymbol{u}_t + \boldsymbol{\epsilon}_t, \tag{18.1}$$

and a linear measurement model:

$$\boldsymbol{z}_t = C_t \boldsymbol{x}_t + \boldsymbol{\delta}_t, \tag{18.2}$$

where $\boldsymbol{x}$ is the state of the system and $\boldsymbol{z}$ are the measurements. Additionally, the Kalman filter assumes the belief distribution of $\boldsymbol{x}$ and the noise terms $\boldsymbol{\epsilon}$, $\boldsymbol{\delta}$ are all Gaussian:

$$bel(\boldsymbol{x}_t) \sim \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t), \quad \boldsymbol{\epsilon}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}_t), \quad \boldsymbol{\delta}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}_t),$$

where $\boldsymbol{R}_t$ and $\boldsymbol{Q}_t$ are the covariances of the state transition and measurement noise models, respectively. With these assumptions the Kalman filter algorithm uses a recursive "predict then correct" approach and the belief will always remain normally distributed.

This algorithm can be used for sensor fusion since the measurement vector $\boldsymbol{z}$ can include measurements from any type of sensor, as long as a linear relationship exists between the measurement and the underlying state $\boldsymbol{x}$ that is to be estimated. At each step of the Kalman filter algorithm, *every* measurement at time t is simultaneously used to update or "correct" the state predicted from the state transition model. Additionally, the Kalman filter takes into account the covariance $\boldsymbol{R}_t$, which includes the covariance of each individual sensor. In fact, the Kalman filter will implicitly favor measurements with lower covariance when performing the correction step[3].

[3] Specifically, this occurs during the computation of the Kalman gain.

A useful trick for applying the Kalman filter to sensor fusion problems is to also note that the state $\boldsymbol{x}$ can contain any type of information, it is not strictly limited to the state usually associated with the robot's dynamics or kinematics. For example, the state could be augmented with auxiliary states such as sensor bias or offsets, or variables to define sensor and actuator health.

Example 18.2.2 (Kalman Filter Multi-Sensor Fusion Example)**.** Consider a self-driving car that has an inertial measurement unit (IMU), a GNSS receiver, and a Lidar unit and where the goal is to leverage all of these sensors to estimate the position, velocity, and acceleration of the vehicle. This suite of sensors can provide noisy position estimates (Lidar and GNSS) as well as noisy acceleration measurements (IMU). For this application, sensor fusion can be accomplished through a Kalman filter.

First, consider a very simple kinematics model that only models longitudinal motion:

$$\dot{p} = v, \quad \dot{v} = a,$$

where p is the longitudinal position, v is the longitudinal velocity, and a is the longitudinal acceleration. This model is then discretized in time by choosing a sampling time T_s, yielding the linear difference equation:

$$\begin{bmatrix} p_{t+1} \\ v_{t+1} \\ a_{t+1} \end{bmatrix} = \begin{bmatrix} 1 & T_s & \frac{T_s^2}{2} \\ 0 & 1 & T_s \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_t \\ v_t \\ a_t \end{bmatrix} + \boldsymbol{\epsilon}_t,$$

where the state is defined as $\boldsymbol{x} = [p, v, a]^\top$, and $\boldsymbol{\epsilon}$ is Gaussian process noise.

It is assumed that the lidar and GNSS sensors directly measure the position p, and that the IMU directly measures the acceleration a, such that the measurement model is:

$$\begin{bmatrix} z_{\text{lidar},t} \\ z_{\text{gnss},t} \\ z_{\text{imu},t} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_t \\ v_t \\ a_t \end{bmatrix} + \boldsymbol{\delta}_t,$$

where $\boldsymbol{\delta}$ is Gaussian measurement noise with zero mean and covariance:

$$\boldsymbol{Q}_t = \begin{bmatrix} \sigma_{\text{lidar}}^2 & 0 & 0 \\ 0 & \sigma_{\text{gnss}}^2 & 0 \\ 0 & 0 & \sigma_{\text{imu}}^2 \end{bmatrix},$$

with $\sigma_{\text{lidar}} = 0.5$, $\sigma_{\text{gnss}} = 0.1$, and $\sigma_{\text{imu}} = 0.2$

Figure 18.2 shows results of the application of the Kalman filter algorithm for fusing these sensor measurements into position estimates. The top plot presents a case where the GNSS sensor is not used, and as can be seen the noisy high-variance lidar measurements result in a noisy estimate of the ground truth position of the car. However, with the addition of the lower-variance GNSS sensor in the bottom figure, the estimate of the position is much more accurate. Generally speaking the estimate would also be more accurate even with the addition of a sensor that was even more noisy than the lidar, but the impact would not be as significant.

18.3 Challenges in Sensor Fusion

Sensor fusion problems can generally be quite challenging, and can vary significantly from application to application. Some of the more common problems in

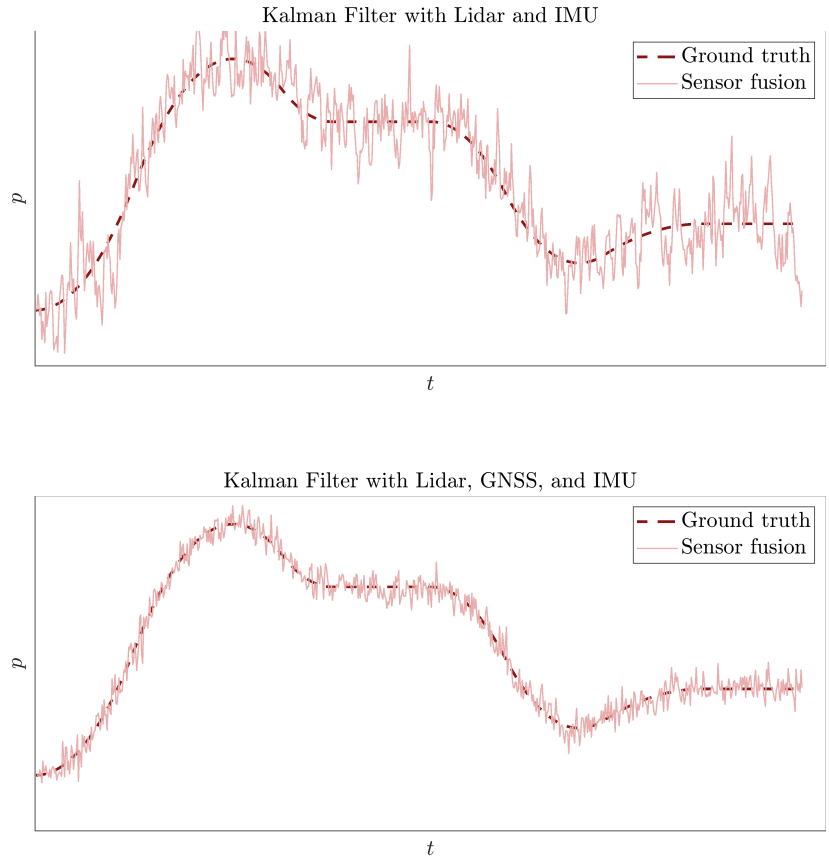

Figure 18.2: Kalman filter sensor fusion for Example 18.2.2. The position of a vehicle is estimated using noisy lidar, GNSS, and IMU data, and the resulting estimate tracks the ground truth. As can be seen, the addition of the lower-variance GNSS results in a better estimate through sensor fusion.

sensor fusion include registration, bias, correlation, data association, and out-of-sequence measurements. The registration problem is that coordinates (both time and space) of different sensors may not always be aligned, which is necessary to ensure they can be appropriately combined. Biases can also arise due to transformations of the data into the unified set of coordinates. Correlation between sensors can also occur, even if they are independently collecting data, and the knowledge of correlation between sensors can have an impact on the best way to fuse the information. In some robotics applications, data association can also be a challenge. One simple example is in multi-target tracking problems, which is similar to the correspondence problem in SLAM problems. Finally, out-of-sequence measurements also pose a logistical challenge in practical sensor fusion applications. These issues often arise due to communication limitations among agents in multi-agent settings. Out-of-sequence measurements might lead to an incorrect temporal order, which in turn causes a negative time measurement update during data fusion fusion. As a consequence, robot localization is biased or a wrong representation of the environment is created. There are a couple of methods to avoid that including external triggering, centralized (time-stamping of data at arrival), or distributed (time-stamping at the time of data acquisition) approaches.

Object Tracking

Autonomous systems, such as self-driving cars or robots, rely critically on an accurate perception of their dynamic environment. Consequently, tracking of other objects, i.e. predicting the state of remote objects given measurement uncertainties, ambiguous measurements, occluded objects, or sensor false alarms, is of great importance for autonomous systems. While single-object tracking (single hypothesis tracking) is well-understood and typically easy to implement (one moving target is tracked by one EKF), the tracking of multiple targets is much more challenging.

18.4 Multi-Object Tracking

Multi-object tracking (MOT) runs a set of estimation filters, one filter for each object to be tracked. While single-model Kalman filters are predominantly in use, a bunch of different approaches exists in the literature such as the interacting multiple model (IMM) filter. Challenges of MOT to be faced are

- inherent uncertainties in prediction (state propagation)
- the data association problem (association of observations and targets)
- track maintenance (creating and deleting tracks)
- multiple reflections from a single object

18.5 Gating

In general, we need to look at every single observation and consider how likely it is to be assigned to a track. To keep the computational efforts low, a screening algorithm (gating) is applied. During this step, observations outside of a specific region for each track are ignored, i.e. the data for assignment is significantly reduced. A rectangular gate is the simplest approach while an ellipsoidal gate is much more intuitive (normal distribution).

18.6 Data Association

Data association or data assignment is the process of linking an observation to a tracked object, i.e. to a track. This can be particularly difficult if we have a large number of targets, many detections, or conflicting hypotheses. Depending on the dimension, we distinguish between 2-D assignment problems (assigning n targets to m observations) and S-D assignment problems (assigning n targets to a set of observations). Within this class, we discuss two typically applied 2-D techniques:

- Global Nearest Neighbor (GNN): is a single hypothesis approach that assigns the global nearest observations to existing tracks and creates new track hypotheses for unassigned observations.
- Joint Probabilistic Data Association (JPDA): is a Bayes-based technique that fuses measurements weighted by the probability of the observation-to-track association. Clustering is usually applied if too many hypotheses are present.

18.7 Track Maintenance

Consists of two steps: deleting a track and creating a track. If a track has not been assigned to a detection at least M times during the last N updates, where N and M are tuning parameters, the track will be deleted. If there is a single unassigned observation, a new tentative track is created. This track is confirmed when detected M times over the last N updates and rejected otherwise.

18.8 Extended Object Tracking

If one moving target generated multiple reflections leading to multiple detections, standard MOT algorithms might fail. This might occur if emerging high-resolution radar sensors are used. These extended objects present new challenges to conventional trackers since those assume a single detection per object per sensor. Extended object tracking (EOT) algorithms are able to deal with this situation. EOT estimate position and velocity, but also the dimensions and the orientation of the moving object. Prominent algorithms are, among others, the Gamma-Gaussian inverse Wishart probability hypothesis density (PHD) tracker and the Gaussian-mixture PHD tracker. Basically, there exist two different but intuitive approaches in MOT:

- Estimating where each individual target is. Each target gets an identity label and targets are tracked while trying to maintain the identities. In situations of targets being closely spaced, that may not be solvable. GNN, JPDA, or multiple hypotheses tracking (MHT) are typically applied along with state estimators.
- Estimating where there are targets. In this case, target identities are not relevant. Typically, a random finite set (RFS) description of the targets is used.

Bibliography

[1] F. Gustafsson. *Statistical Sensor Fusion*. Studentlitteratur, 2013, p. 554.

[2] D. Simon. *Optimal State Estimation: Kalman, H_∞, and Nonlinear Approaches*. John Wiley & Sons, 2006.