

Topics on Robotics & Vision

D. T. McGuiness, PhD

**B.Sc
Mobile Robotics
Lecture Book**

2025.SS



Contents

1 Perception	3
1.1 Introduction	3
1.1.1 Sensors for Mobile Robotics	3
1.1.2 Sensor Classification	4
1.1.3 Characterising Sensor Performance	4
1.1.4 Wheel and Motor Sensors	9
1.2 Active Ranging	16
1.2.1 The Ultrasonic Sensor	17
1.2.2 Motion and Speed Sensors	24
1.3 Vision Based Sensors	27
1.3.1 CMOS Technology	31
1.3.2 Visual Ranging Sensors	32
1.3.3 Depth from Focus	33
1.4 Feature Extraction	37
1.4.1 Defining Feature	37
1.4.2 Using Range Data	39
2 Mobile Robot Localisation	41
2.1 Introduction	41
2.2 The problems of Noise and Aliasing	43
2.2.1 Sensor Noise	43
2.2.2 Sensor Aliasing	44
2.2.3 Effector Noise	45
2.3 Localisation v. Hard-Coded Navigation	48
2.4 Representing Belief	51
2.4.1 Single Hypothesis Belief	51
2.4.2 Multiple Hypothesis Belief	53
2.5 Representing Maps	55
2.5.1 Continuous Representation	56
2.5.2 Decomposition Methods	58
2.5.3 Current Challenges	61
2.6 Probabilistic Map-Based Localisation	64
2.6.1 Introduction	64
2.6.2 Markov Localisation	66
2.6.3 Kalman Filter Localisation	72

2.7	Other Examples of Localisation Methods	77
2.7.1	Landmark-based Navigation	77
2.7.2	Globally Unique Localisation	78
2.7.3	Positioning Beacon systems	80
2.7.4	Route-Based Localisation	81
2.8	Building Maps	82
2.8.1	Stochastic Map Technique	83
2.8.2	Other Mapping Techniques	85
	Bibliography	89

List of Figures

1.1	An example of a rotary encoder. [1]	10
1.2	An example of an electronic compass [2].	11
1.3	Optical Gyroscopes have no moving parts, (unlike mechanical gyroscopes) making them extremely reliable [3].	12
1.4	13
1.5	Signals of an ultrasonic sensor.	17
1.6	An example of an ultrasonic sensor used in Raspberry Pi applications [4].	18
1.8	Schematic of laser rangefinding by phase-shift measurement.	20
1.7	A laser range finder used in robotics applications	20
1.9	Range estimation by measuring the phase shift between transmitted and received signals.	21
1.10	Principle of 1D laser triangulation.	22
1.11	Structured light sources on display at the 2014 Machine Vision Show in Boston [5].	23
1.12	a) Principle of active two dimensional triangulation b) Other possible light structures c) One-dimensional schematic of the principle	23
1.13	Doppler effect between two moving objects (a) or a moving and a stationary object(b)	25
1.14	Sony ICX493AQ/A 10.14-megapixel APS-C (23.4 × 15.6 mm) Charge Coupled Device (CCD) from digital camera Sony DSLR-A200 or DSLR-A300, sensor side [6].	27
1.15	Normalized Spectral Response of a Typical Monochrome CCD.	28
1.16	Types of colour filter used in commercial and industrial applications	28
1.17	Example of white balance. Here the same scene is emulated to be shot under different light conditions [9].	30
1.18	A close-up view of a Complimentary MOS (CMOS) sensor and its circuitry [13].	31
1.19	Photon noise simulation. Number of photons per pixel increases from left to right and from upper row to bottom row [16].	32
1.20	Depiction of the camera optics and its impact on the image. To get a sharp image, the image plane must coincide with the focal plane. Otherwise the image of the point (x, y, z) will be blurred in the image as can be seen in the drawing above.	33
1.21	Three images of the same scene taken with a camera at three different focusing positions. Note the significant change in texture sharpness between the near surface and far surface [17].	34
2.1	Navigation is one if not the most demanding and complicated task in Autonomous Mobile Robotics (AMR). However a successful implementation will result in a versatile AMR which can find its way in unknown environments such as exploring other planets [18].	41

2.2	General schematic for mobile robot localisation.	42
2.3	A sample environment.	48
2.4	An Architecture for Behavior-based Navigation	49
2.5	An Architecture for Map-based (or model-based) Navigation	49
2.6	The three (3) examples of single hypotheses of position using different map representation. a) real map with walls, doors and furniture b) line-based map -> around 100 lines with two parameters c) occupancy grid based map -> around 3000 grid cells sizing 50x50 cm d) topological map using line features (Z/S-lines) and doors -> around 50 features and 18 nodes	52
2.7	The presented robot-centric mapping framework enables mobile robots to create consistent elevation maps of the terrain. Mapping does not necessarily need to be done only in 2D as robots which will be used in outdoor environment would need the height of the map as well [35].	55
2.8	A continuous representation using polygons as environmental obstacles.	56
2.9	Example of a continuous-valued line representation of EPFL. left: real map right: representation with a set of infinite lines.	57
2.10	The schematic for the Kalman filter localisation	75
2.11	An illustration showing the object-level landmarks in blue-boxes. (a,b) shows two different indoor scenarios. The blue boxes represent the 3D object detection of object-level landmarks. The red dots indicate the nodes of the topological map. The yellow lines indicate the edges of the topological map. The green curve is the feasible navigation trajectory generated based on the proposed method [41].	78
2.12	80
2.13	2005 DARPA Grand Challenge winner Stanley performed SLAM as part of its autonomous driving system [49].	82
2.14	General schematic for concurrent localization and map building.	83

List of Tables

2.1 The certainty matrix for the robot [40].	70
------------------------------------------------------	----

List of Examples

List of Theorems

Chapter 1

Perception

Table of Contents

1.1	Introduction	3
1.2	Active Ranging	16
1.3	Vision Based Sensors	27
1.4	Feature Extraction	37

1.1 Introduction

One of the most important tasks of an AMR is to acquire knowledge about its environment.¹ This is achieved by taking measurements using various sensors and then extracting meaningful information from those measurements.

In this chapter we present the most common sensors used in AMR and then discuss strategies for extracting information from the sensors.

¹One could even argue it is the definition of life, if you ask a biologist as the ability to feel and act on its environment is the bare necessity.

1.1.1 Sensors for Mobile Robotics

There is a wide variety of sensors used in AMRs (Fig. 4.1). Some are used to measure simple values like the internal temperature of a robot's electronics or the rotational speed of the motors in its wheels or actuators. Other, more sophisticated sensors can be used to acquire information about the robot's environment or even to directly measure a robot's global position. Here, we focus primarily on sensors used to extract information about the robot's environment. Because a AMR moves around, it will frequently encounter unforeseen environmental characteristics, and therefore such sensing is particularly critical. We begin with a functional classification of sensors. Then, after presenting basic tools for describing a sensor's performance, we proceed to describe selected sensors

in detail.

1.1.2 Sensor Classification

We classify sensors using two (2) important functional axes. Let's define these terms for clarity;

Proprioceptive sensors which measure values **internal** to the robot.

e.g., motor speed, wheel load, robot arm joint angles, battery voltage.

Exteroceptive sensors which measure information from the **robot's environment**;

e.g., distance measurements, light intensity, sound amplitude.

exteroceptive sensor measurements are interpreted by the robot to extract meaningful environmental features.

Passive sensors measure ambient environmental energy entering the sensor.

e.g., temperature probes, microphones and CCD or CMOS cameras.

Active sensors emit energy into the environment, then measure the environmental reaction. Because active sensors can manage more controlled interactions with the environment, they often achieve superior performance. However, active sensing introduces several risks: the outbound energy may affect the very characteristics that the sensor is attempting to measure. Furthermore, an active sensor may suffer from interference between its signal and those beyond its control. For example, signals emitted by other nearby robots, or similar sensors on the same robot may influence the resulting measurements. Examples of active sensors include wheel quadrature encoders, ultrasonic sensors and laser rangefinders.

The sensor classes in Table (4.1) are arranged in ascending order of complexity and descending order of technological maturity. Tactile sensors and proprioceptive sensors are critical to virtually all mobile robots, and are well understood and easily implemented. Commercial quadrature encoders, for example, may be purchased as part of a gear-motor assembly used in a AMR. At the other extreme, visual interpretation by means of one or more CCD/CMOS cameras provides a broad array of potential functionalities, from obstacle avoidance and localisation to human face recognition. However, commercially available sensor units that provide visual functionalities are only now beginning to emerge

1.1.3 Characterising Sensor Performance

The sensors we describe in this chapter vary greatly in their performance characteristics. Some sensors provide extreme accuracy in well-controlled laboratory settings, but are overcome with error

when subjected to real-world environmental variations. Other sensors provide narrow, high precision data in a wide variety settings. To quantify such performance characteristics, first we formally define the sensor performance terminology that will be valuable throughout the rest of this chapter.

Basic Sensor Response Ratings

A number of sensor characteristics can be rated **quantitatively** in a laboratory setting. Such performance ratings will necessarily be best-case scenarios when the sensor is placed on a real-world robot, but are nevertheless useful.

Dynamic Range Used to measure the spread between the lower and upper limits of inputs values to the sensor while maintaining normal sensor operation. Formally, the dynamic range is the ratio of the maximum input value to the minimum measurable input value. Because this raw ratio can be unwieldy, it is usually measured in Decibels, which is computed as ten times the common logarithm of the dynamic range. However, there is potential confusion in the calculation of Decibels, which are meant to measure the ratio between powers, such as Watts or Horsepower.

Suppose your sensor measures motor current and can register values from a minimum of 1 mA to 20 A. The dynamic range of this current sensor is defined as:

$$10 \cdot \log \left[\frac{20}{0.001} \right] = 43 \text{ dB} \quad (1.1)$$

Now suppose you have a voltage sensor that measures the voltage of your robot's battery, measuring any value from 1 mV to 20 V. Voltage is **NOT** a unit of power, but the square of voltage is proportional to power. Therefore, we use 20 instead of 10:

$$20 \cdot \log \left[\frac{20}{0.001} \right] = 86 \text{ dB} \quad (1.2)$$

Range An important rating in AMR because often robot sensors operate in environments where they are frequently exposed to input values beyond their working range. In such cases, it is critical to understand how the sensor will respond. For example, an optical rangefinder will have a minimum operating range and can thus provide spurious data when measurements are taken with object closer than that minimum.

Resolution The minimum difference between two (2) values that can be detected by a sensor. Usually, the lower limit of the dynamic range of a sensor is equal to its resolution. However, in the case of digital sensors, this is not necessarily so. For example, suppose that you have a sensor that measures voltage, performs an analogue-to-digital conversion and outputs the converted value as an 8-bit number linearly corresponding to between 0 and 5 Volts. If this sensor is truly linear, then it has $2^8 - 1$ total output values or a resolution of:

$$\frac{5}{255} = 20 \text{ mV}$$

Linearity is an important measure governing the behaviour of the sensor's output signal as the input signal varies. A linear response indicates that if two (2) inputs, say x and y result in the two outputs $f(x)$ and $f(y)$, then for any values a and b , the following relation can be derived:

$$f(x + y) = f(x) + f(y).$$

This means that a plot of the sensor's input/output response is simply a straight line.

Bandwidth or Frequency is used to measure the speed with which a sensor can provide a stream of readings. Formally, the number of measurements per second is defined as the sensor's frequency in Hz. Because of the dynamics of moving through their environment, mobile robots often are limited in maximum speed by the bandwidth of their obstacle detection sensors. Thus increasing the bandwidth of ranging and vision-based sensors has been a high-priority goal in the robotics community.

In Situ Sensor Performance

The above sensor characteristics can be reasonably measured in a laboratory environment, with confident extrapolation to performance in real-world deployment. However, a number of important measures cannot be reliably acquired without deep understanding of the complex interaction between all environmental characteristics and the sensors in question. This is most relevant to the most sophisticated sensors, including active ranging sensors and visual interpretation sensors.

Sensitivity A measure of the degree to which an incremental change in the target input signal changes the output signal. Formally, sensitivity is the ratio of output change to input change. Unfortunately, however, the sensitivity of exteroceptive sensors is often confounded by undesirable sensitivity and performance coupling to other environmental parameters.

Cross-Sensitivity is the technical term for sensitivity to environmental parameters that are orthogonal to the target parameters for the sensor. For example, a flux-gate compass can demonstrate high sensitivity to magnetic north and is therefore of use for AMR navigation. However, the compass will also demonstrate high sensitivity to ferrous building materials, so much so that its cross-sensitivity often makes the sensor useless in some indoor environments. High cross-sensitivity of a sensor is generally undesirable, especially so when it cannot be modelled.

Error of a sensor is defined as the difference between the sensor's output measurements and the true values being measured, within some specific operating context.

As an example, given a true value v and a measured value m , we can define error as:

$$\text{Error} = m - v.$$

Accuracy defined as the degree of conformity between the sensor's measurement and the true value, and is often expressed as a proportion of the true value (e.g. 97.5% accuracy):

$$\text{Accuracy} = 1 - \frac{|m - v|}{v}.$$

Of course, obtaining the ground truth (v), can be difficult or impossible, and so establishing a confident characterisation of sensor accuracy can be problematic. Further, it is important to distinguish between two different sources of error:

- Systematic errors are caused by factors or processes that can in theory be modelled. These errors are, therefore, deterministic.²

Poor calibration of a laser rangefinder, un-modelled slope of a hallway floor and a bent stereo camera head due to an earlier collision are all possible causes of systematic sensor errors

²Meaning, its value is not determined by a random process and therefore should, in theory, be predictable.

- Random errors cannot be predicted using a sophisticated model nor can they be mitigated with more precise sensor machinery. These errors can only be described in probabilistic terms (i.e. stochastic). Hue instability in a colour camera, spurious range-finding errors and black level noise in a camera are all examples of random errors.

Precision is often confused with accuracy, and now we have the tools to clearly distinguish these two terms. Intuitively, high precision relates to reproducibility of the sensor results. For example, one sensor taking multiple readings of the same environmental state has high precision if it produces the same output. In another example, multiple copies of this sensor taking readings of the same environmental state have high precision if their outputs agree. Precision does not, however, have any bearing on the accuracy of the sensor's output with respect to the true value being measured. Suppose that the random error of a sensor is characterised by some mean value (μ) and a standard deviation (σ). The formal definition of precision is the ratio of the sensor's output range to the standard deviation:

$$\text{Precision} = \frac{\text{Range}}{\sigma}.$$

Only σ and **NOT** μ has impact on precision. In contrast mean error is directly proportional to overall sensor error and inversely proportional to sensor accuracy.

Characterising Error

Mobile robots depend heavily on **exteroceptive** sensors. Many of these sensors concentrate on a central task for the robot:

acquiring information on objects in the robot's immediate vicinity so that it may interpret the state of its surroundings.

Of course, these "objects" surrounding the robot are all detected from the viewpoint of its local reference frame.³ Since the systems we study are **mobile**, their ever-changing position and their motion has a significant impact on overall sensor behaviour.

³In this case we are referring to the robot reference frame.

Now that we have the necessary knowledge on the fundamental concepts and terminology, we can

now describe how dramatically the sensor error of an AMR **disagrees** with the ideal picture drawn in the previous section.

Blurring of Systematical and Random Errors

Active ranging sensors tend to have failure modes which are triggered largely by specific relative positions of the sensor and environment targets.

⁴The incident light is reflected into a single outgoing direction.

For example, a sonar sensor will product specular reflections,⁴ producing grossly inaccurate measurements of range, at specific angles to a smooth sheet-rock wall.

During motion of the robot, such relative angles occur at stochastic intervals. This is especially true in a AMR outfitted with a ring of multiple sonars. The chances of one sonar entering this error mode during robot motion is high. From the perspective of the moving robot, the sonar measurement error is a **random error** in this case. However, if the robot were to stop, becoming motionless, then a very different error modality is possible.

If the robot's static position causes a particular sonar to fail in this manner, the sonar will fail consistently and will tend to return precisely the same (and incorrect!) reading time after time. Once the robot is motionless, the error appears to be systematic and high precision.

The fundamental mechanism at work here is the cross-sensitivity of AMR sensors to robot pose and robot-environment dynamics.

The models for such cross-sensitivity are **NOT**, in an underlying sense, truly random. However, these physical interrelationships are rarely modelled and therefore, from the point of view of an incomplete model, the errors appear random during motion and systematic when the robot is at rest. Sonar is not the only sensor subject to this blurring of systematic and random error modality. Visual interpretation through the use of a CCD camera is also highly susceptible to robot motion and position because of camera dependency on lighting.⁵

⁵such as glare and reflections.

The important point is to realise that, while systematic error and random error are well-defined in a controlled setting, the AMR can exhibit error characteristics that bridge the gap between deterministic and stochastic error mechanisms.

Multi-Modal Error Distributions

It is common to characterise the behaviour of a sensor's random error in terms of a probability distribution over various output values. In general, one knows very little about the causes of random error and therefore several simplifying assumptions are commonly used. For example, we can assume that the error is zero-mean ($\mu = 0$), in that it symmetrically generates both positive and negative measurement error. We can go even further and assume that the probability density curve is Gaussian.

Although we discuss the mathematics of this in detail later, it is important for now to recognise the fact that one frequently assumes symmetry as well as unimodal distribution. This means that measuring the correct value is most probable, and any measurement that is further away from the correct value is less likely than any measurement that is closer to the correct value. These are strong assumptions that enable powerful mathematical principles to be applied to AMR problems, but it is important to realise how wrong these assumptions usually are.

Consider, for example, the sonar sensor once again. When ranging an object that reflects the sound signal well, the sonar will exhibit high accuracy, and will induce random error based on noise, for example, in the timing circuitry. This portion of its sensor behaviour will exhibit error characteristics that are fairly **symmetric** and **unimodal**. However, when the sonar sensor is moving through an environment and is sometimes faced with materials that cause coherent reflection rather than returning the sound signal to the sonar sensor, then the sonar will grossly overestimate distance to the object. In such cases, the error will be biased toward positive measurement error and will be far from the correct value. The error is not strictly systematic, and so we are left modelling it as a probability distribution of random error. So the sonar sensor has two (2) separate types of operational modes, one in which the signal does return and some random error is possible, and the second in which the signal returns after a multi-path reflection, and gross overestimation error occurs. The probability distribution could easily be at least bimodal in this case, and since overestimation is more common than underestimation it will also be asymmetric.

As a second example, consider ranging via stereo vision. Once again, we can identify two (2) modes of operation. If the stereo vision system correctly correlates two images, then the resulting random error will be caused by camera noise and will limit the measurement accuracy. But the stereo vision system can also correlate two images incorrectly, matching two fence posts for example that are not the same post in the real world. In such a case stereo vision will exhibit gross measurement error, and one can easily imagine such behaviour violating both the unimodal and the symmetric assumptions. The thesis of this section is that sensors in a AMR may be subject to multiple modes of operation and, when the sensor error is characterised, uni modality and symmetry may be grossly violated. Nonetheless, as you will see, many successful AMR systems make use of these simplifying assumptions and the resulting mathematical techniques with great empirical success. The above sections have presented a terminology with which we can characterise the advantages and disadvantages of various mobile robot sensors. In the following sections, we do the same for a sampling of the most commonly used AMR sensors today.

1.1.4 Wheel and Motor Sensors

Wheel/motor sensors are devices used to measure the internal state and dynamics of a mobile robot. These sensors have vast applications outside of AMR and, as a result, AMR has enjoyed the benefits of high-quality, low-cost wheel and motor sensors which offer excellent resolution.

In the next part, we sample just one such sensor, the optical incremental encoder.

Optical Encoders

Optical incremental encoders have become the most popular device for measuring angular speed and position within a motor drive or at the shaft of a wheel or steering mechanism. In mobile robotics, encoders are used to control the position or speed of wheels and other motor-driven joints. Because these sensors are proprioceptive, their estimate of position is best in the reference frame of the robot and, when applied to the problem of robot localisation, significant corrections are required as discussed in Chapter 5.

An optical encoder is basically a mechanical light chopper that produces a certain number of sine or square wave pulses for each shaft revolution. It consists of an illumination source, a fixed grating that masks the light, a rotor disc with a fine optical grid that rotates with the shaft, and fixed optical detectors. As the rotor moves, the amount of light striking the optical detectors varies based on the alignment of the fixed and moving gratings. In robotics, the resulting sine wave is transformed into a discrete square wave using a threshold to choose between light and dark states. Resolution is measured in Cycles Per Revolution (CPR).

The minimum angular resolution can be readily computed from an encoder's CPR rating. A typical encoder in AMR may have 2,000 CPR while the optical encoder industry can readily manufacture encoders with 10,000 CPR. In terms of required bandwidth, it is of course critical that the encoder be sufficiently fast to count at the shaft spin speeds that are expected. Industrial optical encoders present no bandwidth limitation to AMR applications. Usually in AMR the quadrature encoder is used. In this case, a second illumination and detector pair is placed 90° shifted with respect to the original in terms of the rotor disc. The resulting twin square waves, shown in Fig. 4.2, provide significantly more information. The ordering of which square wave produces a rising edge first identifies the direction of rotation. Furthermore, the four detectability different states improve the resolution by a factor of four with no change to the rotor disc. Thus, a 2,000 CPR encoder in quadrature yields 8,000 counts. Further improvement is possible by retaining the sinusoidal wave measured by the optical detectors and performing sophisticated interpolation. Such methods, although rare in AMR, can yield 1000-fold improvements in resolution. As with most proprioceptive sensors, encoders are generally in the controlled environment of a AMR's internal structure, and so systematic error and cross-sensitivity can be engineered away. The accuracy of optical encoders is often assumed to be 100% and, although this may not entirely correct, any errors at the level of an optical encoder are dwarfed by errors downstream of the motor shaft.



Figure 1.1: An example of a rotary encoder. [1]

Heading Sensors

Heading sensors can be proprioceptive (gyroscope, inclinometer) or exteroceptive (compass). They are used to determine the robots orientation and inclination. They allow us, together with appropriate velocity information, to integrate the movement to a position estimate. This procedure,

which has its roots in vessel and ship navigation, is called dead reckoning.

Compasses

The two most common modern sensors for measuring the direction of a magnetic field are the Hall Effect and Flux Gate compasses. Each has advantages and disadvantages, as described below. The Hall Effect describes the behaviour of electric potential in a semiconductor when in the presence of a magnetic field. When a constant current is applied across the length of a semi-conductor, there will be a voltage difference in the perpendicular direction, across the semi-conductor's width, based on the relative orientation of the semiconductor to magnetic flux

lines. In addition, the sign of the voltage potential identifies the direction of the magnetic field. Thus, a single semiconductor provides a measurement of flux and direction along one dimension. Hall Effect digital compasses are popular in AMR, and contain two such semiconductors at right angles, providing two axes of magnetic field (thresholded) direction, thereby yielding one of 8 possible compass directions. The instruments are inexpensive but also suffer from a range of disadvantages. Resolution of a digital hall effect compass is poor. Internal sources of error include the nonlinearity of the basic sensor and systematic bias errors at the semiconductor level. The resulting circuitry must perform significant filtering, and this lowers the bandwidth of hall effect compasses to values that are slow in AMR terms. For example the hall effect compasses pictured in figure 4.3 needs 2.5 seconds to settle after a 90° spin. The Flux Gate compass operates on a different principle. Two small coils are wound on fer- rite cores and are fixed perpendicular to one-another. When alternating current is activated in both coils, the magnetic field causes shifts in the phase depending upon its relative alignment with each coil. By measuring both phase shifts, the direction of the magnetic field in two dimensions can be computed. The flux-gate compass can accurately measure the strength of a magnetic field and has improved resolution and accuracy; however it is both larger and more expensive than a Hall Effect compass. Regardless of the type of compass used, a major drawback concerning the use of the Earth's magnetic field for AMR applications involves disturbance of that magnetic field by other magnetic objects and man-made structures, as well as the bandwidth limitations of electronic compasses and their susceptibility to vibration. Particularly in indoor environments AMR applications have often avoided the use of compasses, although a compass can conceivably provide useful local orientation information indoors, even in the presence of steel structures.



Figure 1.2: An example of an electronic compass [2].

Gyroscope

Gyroscopes are heading sensors which preserve their orientation in relation to a fixed reference frame. Thus they provide an absolute measure for the heading of a mobile system. Gyroscopes

can be classified in two categories, mechanical gyroscopes and optical gyroscopes.

Mechanical Gyroscopes

The concept of a mechanical gyroscope relies on the inertial properties of a fast spinning rotor. The property of interest is known as the gyroscopic precession. If you try to rotate a fast spinning wheel around its vertical axis, you will feel a harsh reaction in the horizontal axis. This is due to the angular momentum associated with a spinning wheel and will keep the axis of the gyroscope inertially stable. The reactive torque τ and thus the tracking stability with the inertial frame are proportional to the spinning speed ω , the precession speed Ω and the wheel's inertia I .

$$\tau = I\omega\Omega$$

By arranging a spinning wheel as seen in Figure 4.4, no torque can be transmitted from the outer pivot to the wheel axis. The spinning axis will therefore be space-stable (i.e. fixed in an inertial reference frame). Nevertheless, the remaining friction in the bearings of the gyro- axis introduce small torques, thus limiting the long term space stability and introducing small errors over time. A high quality mechanical gyroscope can cost up to \$100,000 and has an angular drift of about 0.1̄ in 6 hours. For navigation, the spinning axis has to be initially selected. If the spinning axis is aligned with the north-south meridian, the earth's rotation has no effect on the gyro's horizontal axis. If it points east-west, the horizontal axis reads the earth rotation. Rate gyros have the same basic arrangement as shown in Figure 4.4 but with a slight modification. The gimbals are restrained by a torsional spring with additional viscous damping. This enables the sensor to measure angular speeds instead of absolute orientation.

Optical Gyroscopes

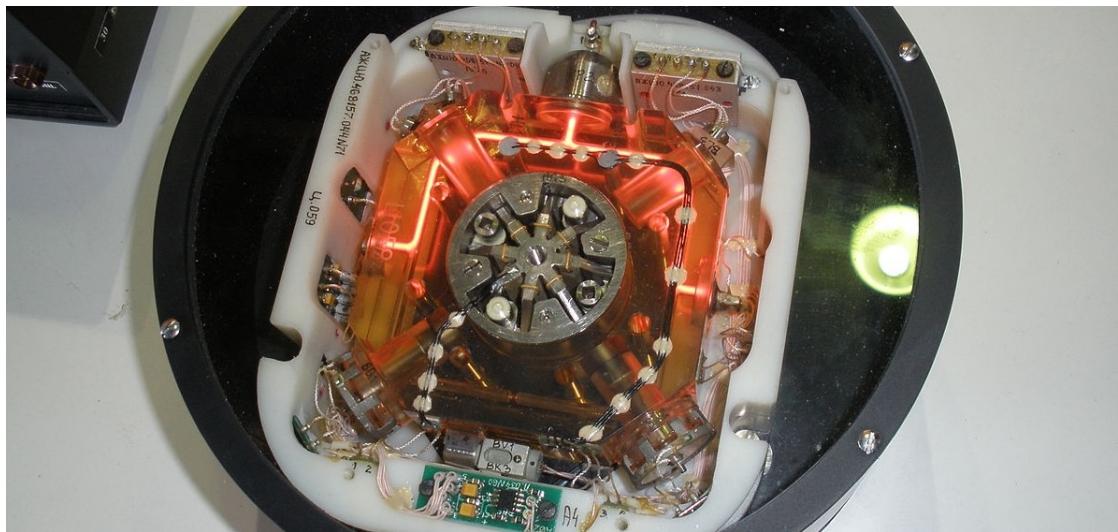


Figure 1.3: Optical Gyroscopes have no moving parts, (unlike mechanical gyroscopes) making them extremely reliable [3].

Optical gyroscopes are a relatively new innovation. Commercial use began in the early 1980's when they were first installed in aircraft. Optical gyroscopes are angular speed sensors that use two monochromatic light beams, or lasers, emitted from the same source instead of moving, mechanical parts. They work on the principle that the speed of light remains unchanged and, therefore, geometric change can cause light to take a varying amount of time to reach its destination. One laser beam is sent traveling clockwise through a fiber while the other travels counterclockwise. Because the laser traveling in the direction of rotation has a slightly shorter path, it will have a higher frequency. The difference in frequency of the two beams is proportional to the angular velocity of the cylinder. New solid-state optical gyroscopes based on the same principle are built using microfabrication technology, thereby providing heading information with resolution and bandwidth far beyond the needs of mobile robotic applications. Bandwidth, for instance, can easily exceed 100KHz while resolution can be smaller than 0.0001°/hr.

Ground Based Beacons



Figure 1.4

One elegant approach to solving the localization problem in AMR is to use active or passive beacons. Using the interaction of on-board sensors and the environmental beacons, the robot can identify its position precisely. Although the general intuition is identical to that of early human navigation beacons, such as stars, mountains and lighthouses, modern technology has enabled sensors to localize an outdoor robot with accuracies of better than 5 cm within areas that are kilometres in size.

In the following subsection, we describe one such beacon system, the Global Positioning System (GPS), which is extremely effective for outdoor ground-based and flying robots. Indoor beacon systems have been generally less successful for a number of reasons. The expense of environmental modification in an indoor setting is not amortized over an extremely large useful area, as it is for example in the case of GPS. Furthermore, indoor environments offer significant challenges not seen outdoors, including multipath and environment dynamics. A laser-based indoor beacon system, for example, must disambiguate the one true laser signal from possibly tens of other powerful signals that have reflected off of walls, smooth floors and doors. Confounding this, humans and other obstacles

may be constantly changing the environment, for example occluding the one true path from the beacon to the robot. In commercial applications such as manufacturing plants, the environment can be carefully controlled to ensure success. In less structured indoor settings, beacons have nonetheless been used, and the problems are mitigated by careful beacon placement and the useful of passive sensing modalities.

Global Positioning System

The Global Positioning System (GPS) was initially developed for military use but is now freely available for civilian navigation. There are at least 24 operational GPS satellites at all times. The satellites orbit every 12 hours at a height of 20.190km. There are four (4) satellites which located in each of six planes inclined 55° with respect to the plane of the earth's equator (figure 4.5).

Each satellite continuously transmits data which indicates its location and the current time. Therefore, GPS receivers are **completely passive** but **exteroceptive** sensors. The GPS satellites synchronise their transmissions to allow their signals to be sent at the same time. When a GPS receiver reads the transmission of two (2) or more satellites, the arrival time differences inform the receiver as to its relative distance to each satellite.

By combining information regarding the arrival time and instantaneous location of four (4) satellites, the receiver can infer its own position.

In theory, such triangulation requires only three (3) data points. However, timing is extremely critical in the GPS application because the time intervals being measured are in ns.

It is, of course, mandatory the satellites to be well synchronised. To this end, they are updated by ground stations regularly and each satellite carries on-board atomic clocks⁶ for timing. The GPS receiver clock is also important so that the travel time of each satellite's transmission can be accurately measured. But GPS receivers have a simple quartz clock. So, although 3 satellites would ideally provide position in three axes, the GPS receiver requires 4 satellites, using the additional information to solve for 4 variables: three position axes plus a time correction. The fact that the GPS receiver must read the transmission of 4 satellites simultaneously is a significant limitation. GPS satellite transmissions are extremely low-power, and reading them successfully requires direct line-of-sight communication with the satellite. Thus, in confined spaces such as city blocks with tall buildings or dense forests, one is unlikely to receive 4 satellites reliably. Of course, most indoor spaces will also fail to provide sufficient visibility of the sky for a GPS receiver to function. For these reasons, GPS has been a popular sensor in AMR, but has been relegated to projects involving AMR traversal of wide-open spaces and autonomous flying machines. A number of factors affect the performance of a localization sensor that makes use of GPS. First, it is important to understand that, because of the specific orbital paths of the GPS satellites, coverage is not geometrically identical in different portions of the Earth and therefore resolution is not uniform. Specifically, at the North and South poles, the satellites are very close to the horizon and, thus, while resolution in the latitude and longitude directions is good, resolution of altitude is relatively poor as compared to



⁶An example of a cesium clock for use in GPS.

more equatorial locations.

The second point is that GPS satellites are merely an information source. They can be employed with various strategies in order to achieve dramatically different levels of localisation resolution. The basic strategy for GPS use, called pseudorange and described above, generally performs at a resolution of 15m. An extension of this method is differential GPS, which makes use of a second receiver that is static and at a known exact position. A number of errors can be corrected using this reference, and so resolution improves to the order of 1m or less. A disadvantage of this technique is that the stationary receiver must be installed, its location must be measured very carefully and of course the moving robot must be within kilometers of this static unit in order to benefit from the DGPS technique. A further improved strategy is to take into account the phase of the carrier signals of each received satellite transmission. There are two carriers, at 19cm and 24cm, therefore significant improvements in precision are possible when the phase difference between multiple satellites is measured successfully. Such receivers can achieve 1cm resolution for point positions and, with the use of multiple receivers as in DGPS, sub-1cm resolution. A final consideration for AMR applications is bandwidth. GPS will generally offer no better than 200 - 300ms latency, and so one can expect no better than 5Hz GPS updates. On a fast-moving AMR or flying robot, this can mean that local motion integration will be required for proper control due to GPS latency limitations.

1.2 Active Ranging

Active range sensors continue to be the most popular sensors used in AMR. Many ranging sensors have a low price point, and most importantly all ranging sensors provide easily interpreted outputs:

Direct measurements of distance from the robot to objects in its vicinity.

For obstacle detection and avoidance, most AMR rely heavily on active ranging sensors. But the local free-space information provided by range sensors can also be accumulated into representations beyond the robot's current local reference frame. Therefore, active range sensors are also commonly found as part of the localisation and environmental modelling processes of AMRs.

It is only with the slow advent of successful visual interpretation competency that we can expect the class of active ranging sensors to gradually lose their primacy as the sensor class of choice among AMR engineers.

Below, we present two (2) Time-of-Flight (ToF) active range sensors:

- the ultrasonic sensor,
- the laser rangefinder.

Continuing onwards, we then present two (2) geometric active range sensors:

- the optical triangulation sensor,
- the structured light sensor.

Time-of-Flight Active Ranging

ToF ranging makes use of the [propagation speed of sound](#) or an [electromagnetic wave](#). In general, the travel distance of a sound or electromagnetic wave is given by:

$$d = ct,$$

where d is the distance travelled usually round-trip (m), c the speed of wave propagation (ms^{-1}), and t is the time it takes to travel (s).

It is important to point out the propagation speed v of sound is approximately 0.3 m ms^{-1} whereas the speed of an electromagnetic signal is 0.3 m ns^{-1} , which is one million times faster. The ToF for a typical distance, say 3 m, is 10 ms for an ultrasonic system but only 10 ns for a laser rangefinder. It is therefore obvious that measuring the time of flight t with electromagnetic signals is more technologically challenging.⁷

The quality of ToF range sensors depends mainly on the following:

⁷This explains why laser range sensors have only recently become affordable and robust for use on mobile robots.

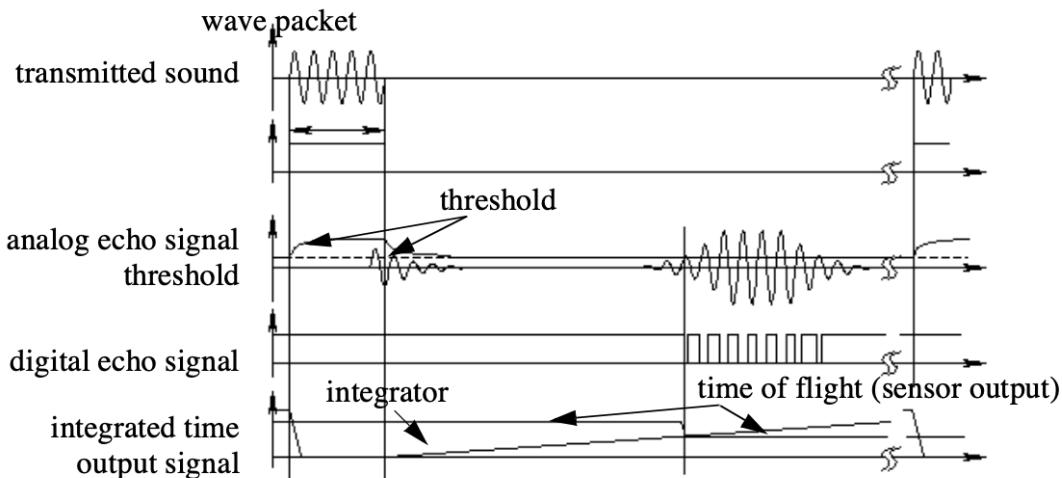


Figure 1.5: Signals of an ultrasonic sensor.

- Uncertainties in determining the exact time of arrival of the reflected signal,
- Inaccuracies in the time of flight measurement, particularly with laser range sensors,
- The dispersal cone of the transmitted beam mainly with ultrasonic range sensors
- Interaction with the target (e.g., surface absorption, specular reflections)
- Variation of propagation speed, and
- The speed of the AMR and target (in the case of a dynamic target).

As discussed below, each type of ToF sensor is sensitive to a particular subset of the above list of factors.

1.2.1 The Ultrasonic Sensor

The main ethos of an ultrasonic⁸ sensor is to transmit a packet of ultrasonic pressure waves and to measure the time it takes for this wave to reflect and return to the receiver. The distance d of the object causing the reflection can be calculated based on the propagation speed of sound⁹ c and the time of flight t .

$$d = \frac{c \times t}{2}$$

The speed of sound (v) in air is given by the following relation:

$$v = \sqrt{\gamma RT}$$

where γ is the ratio of specific heat, R is the gas constant ($\text{J mol}^{-1} \text{K}^{-1}$), and T is the temperature

⁸Ultrasound is sound with frequencies greater than 20 kHz.

⁹Of course in this regard careful consideration needs to be made if the medium is significantly different than that of air (i.e., water).

in Kelvin (K). In air, at standard pressure, and 20 °C the speed of sound is approximately:

$$v = 343 \text{ m s}^{-1}.$$

We can see the different signal output and input of an ultrasonic sensor in **Fig. 1.5**.

First, a series of sound pulses are emitted, which creates the wave packet. An integrator also begins to **linearly climb** in value, measuring the time from the transmission of these sound waves to detection of an echo. A threshold value is set for triggering an incoming sound wave as a valid echo.

This threshold is often decreasing in time, because the amplitude of the expected echo decreases over time based on dispersal as it travels longer.

But during transmission of the initial sound pulses and just afterwards, the threshold is set very high to suppress triggering the echo detector with the outgoing sound pulses. A transducer will continue to ring for up to several ms after the initial transmission, and this governs the blanking time of the sensor.

If, during the blanking time, the transmitted sound were to reflect off of an extremely close object and return to the ultrasonic sensor, it may fail to be detected.

However, once the blanking interval has passed, the system will detect any above-threshold reflected sound, triggering a digital signal and producing the distance measurement using the integrator value.

The ultrasonic wave typically has a frequency between 40 and 180 kHz and is usually generated by a piezo or electrostatic transducer. Often the same unit is used to measure the reflected signal, although the required blanking interval can be reduced through the use of separate output and input devices. Frequency can be used to select a useful range when choosing the appropriate ultrasonic sensor for a AMR. Lower frequencies correspond to a longer range, but with the disadvantage of longer post-transmission ringing and, therefore, the need for longer blanking intervals.

Most ultrasonic sensors used by AMRs have an effective range of roughly 12 cm to 5 metres. The published accuracy of commercial ultrasonic sensors varies between 98% and 99.1%. In AMR applications, specific implementations generally achieve a resolution of approximately 2 cm.

In most cases one may want a narrow opening angle for the sound beam in order to also obtain precise directional information about objects that are encountered. This is a major limitation since sound propagates in a cone-like manner with opening angles around 20° and 40°. Consequently, when using ultrasonic ranging one does not acquire depth data points but, rather, entire regions of constant depth. This means that the sensor tells us only that there is an object at a certain distance in within the area of the measurement cone. The sensor readings must be plotted as segments of an arc (sphere for 3D) and not as point measurements.¹⁰ However, recent research developments

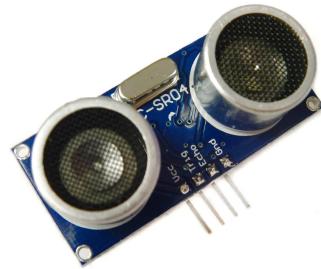
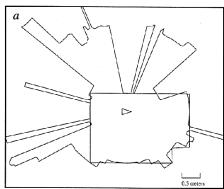


Figure 1.6: An example of an ultrasonic sensor used in Raspberry Pi applications [4].



¹⁰The results of a 360° scan of a room.

show significant improvement of the measurement quality in using sophisticated echo processing. Ultrasonic sensors suffer from several additional drawbacks, namely in the areas of **error**, **bandwidth** and **cross-sensitivity**. The published accuracy values for ultrasonic sensors are nominal values based on successful, perpendicular reflections of the sound wave off an acoustically reflective material.

This does not capture the effective error modality seen on a AMR moving through its environment. As the ultrasonic transducer's angle to the object being ranged varies away from perpendicular, the chances become good that the sound waves will coherently reflect away from the sensor, just as light at a shallow angle reflects off of a mirror. Therefore, the true error behavior of ultrasonic sensors is compound, with a well-understood error distribution near the true value in the case of a successful retro-reflection, and a more poorly-understood set of range values that are grossly larger than the true value in the case of coherent reflection.

Of course the acoustic properties of the material being ranged have direct impact on the sensor's performance. Again, the impact is discrete, with one material possibly failing to produce a reflection that is sufficiently strong to be sensed by the unit. For example, foam, fur and cloth can, in various circumstances, acoustically absorb the sound waves. A final limitation for ultrasonic ranging relates to bandwidth. Particularly in moderately open spaces, a single ultrasonic sensor has a relatively slow cycle time.

For example, measuring the distance to an object that is 3 m away will take such a sensor 20ms, limiting its operating speed to 50 Hz. But if the robot has a ring of 20 ultrasonic sensors, each firing sequentially and measuring to minimize interference between the sensors, then the ring's cycle time becomes 0.4s and the overall update frequency of any one sensor is just 2.5 Hz. For a robot conducting moderate speed motion while avoiding obstacles using ultrasonic sensor, this update rate can have a measurable impact on the maximum speed possible while still sensing and avoiding obstacles safely.

Ultrasonic measurements may be limited through barrier layers with large salinity, temperature or vortex differentials.

Laser Rangefinder

The laser rangefinder is a ToF sensor which achieves significant improvements over the ultrasonic range sensor due to the **use of laser light instead of sound**. This type of sensor consists of a transmitter which illuminates a target with a collimated¹¹ beam (e.g. laser), and a receiver capable of detecting the component of light which is essentially coaxial with the transmitted beam. Often referred to as optical radar or Light Detection and Ranging (LIDAR), these devices produce a range estimate based on the time needed for the light to reach the target and return.

¹¹meaning all the rays in questions are made accurately parallel.

A mechanical mechanism with a mirror sweeps the light beam to cover the required scene in a plane or even in 3 dimensions, using a rotating mirror. One way to measure the ToF for the light beam is to use a pulsed laser and then measured the elapsed time directly, just as in the ultrasonic solution

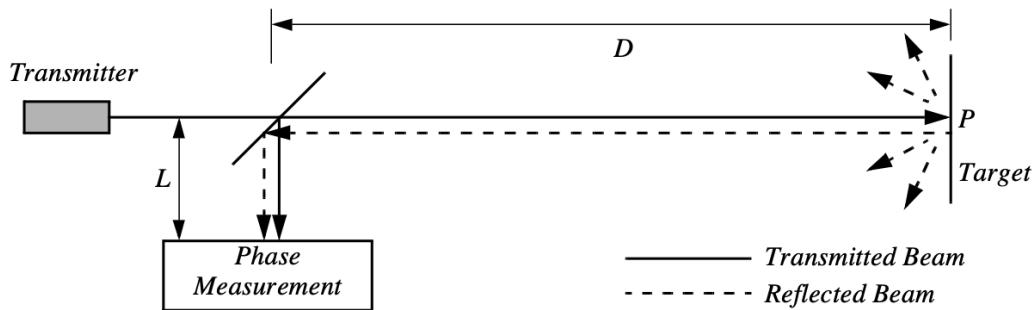


Figure 1.8: Schematic of laser rangefinding by phase-shift measurement.

described in just a little bit. Electronics capable of resolving ps are required in such devices and they are therefore very expensive. A second method is to measure the beat frequency between a frequency modulated continuous wave and its received reflection. Another, even easier method is to measure the phase shift of the reflected light.

Continuous Wave Radar It is a type of radar system where a known stable frequency continuous wave radio energy is transmitted and then received from any reflecting objects. Individual objects can be detected using the Doppler effect, which causes the received signal to have a different frequency from the transmitted signal, allowing it to be detected by filtering out the transmitted frequency.

Doppler-analysis of radar returns can allow the filtering out of slow or non-moving objects, thus offering immunity to interference from large stationary objects and slow-moving clutter. This makes it particularly useful for looking for objects against a background reflector, for instance, allowing a high-flying aircraft to look for aircraft flying at low altitudes against the background of the surface. Because the very strong reflection off the surface can be filtered out, the much smaller reflection from a target can still be seen.



Figure 1.7: A laser range finder used in robotics applications

Phase Shift Measurement Near infrared light, which could be from an Light-Emitting Diode (LED) or a laser, is collimated and transmitted from the transmitter T in Fig. 1.8 and hits a point P in the environment.

For surfaces having a roughness greater than the wavelength of the incident light, diffuse reflection will occur, meaning that the light is reflected almost isotropically¹². The wavelength of the infrared light emitted is 824 nm and so most surfaces with the exception of only highly polished reflecting objects, will be diffuse reflectors. The component of the infrared light which falls within the receiving aperture of the sensor will return almost parallel to the transmitted beam, for distant objects. The sensor transmits 100% amplitude modulated light at a known frequency and measures the phase

¹²Something that is isotropic has the same size or physical properties when it is measured in different directions

shift between the transmitted and reflected signals.

Fig. 1.9 shows how this technique can be used to measure range. The wavelength of the modulating signal obeys the equation $c = f\lambda$ where c is the speed of light and f the modulating frequency.

For example, $f = 5 \text{ MHz}$, the wavelength is $\lambda = 60 \text{ m}$.

The total distance D' covered by the emitted light is:

$$D' = L + 2D = L \frac{\theta}{2\pi} \lambda$$

where D and L are the distances defined in **Fig. 1.8**. The required distance D , between the beam splitter and the target, is therefore given by:

$$D = \frac{\lambda}{4\pi} \theta$$

where θ is the electronically measured phase difference between the transmitted and reflected light beams, and λ the known modulating wavelength. It can be seen that the transmission of a single frequency modulated wave can theoretically result in ambiguous range estimates since

For example if $\lambda = 60\text{m}$, a target at a range of 5 m would give an indistinguishable phase measurement from a target at 65 m , since each phase angle would be 360° apart.

We therefore define an **ambiguity interval** of λ , but in practice we note that the range of the sensor is much lower than λ due to the attenuation of the signal in air. It can be shown that the confidence in the range (phase estimate) is inversely proportional to the square of the received signal amplitude, directly affecting the sensor's accuracy. Hence dark, distant objects will not produce as good range estimates as close, bright objects.

As with ultrasonic ranging sensors, an important error mode involves coherent reflection of the energy. With light, this will only occur when striking a highly polished surface. Practically, a AMR may encounter such surfaces in the form of a polished desktop, file cabinet or of course a mirror. Unlike ultrasonic sensors, laser rangefinders cannot detect the presence of optically transparent materials such as glass, and this can be a significant obstacle in environments, for example museums, where glass is commonly used.

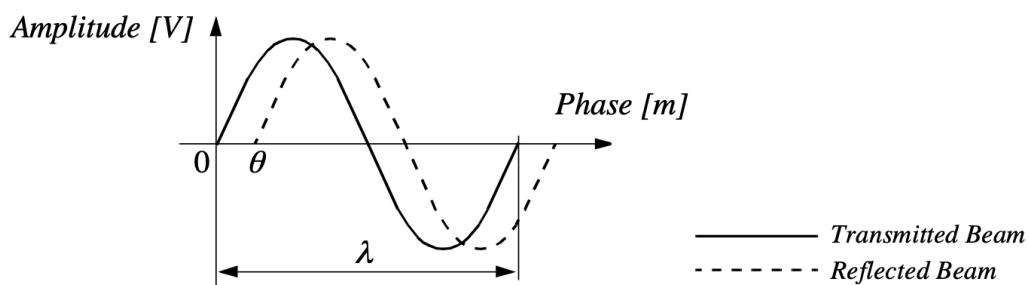


Figure 1.9: Range estimation by measuring the phase shift between transmitted and received signals.

Triangulation-based Active Ranging

Triangulation-based ranging sensors use geometrical properties in their measuring strategy to establish distance readings to objects. The simplest class of triangulation-based rangers are active because they project a known light pattern (e.g., a point, a line or a texture) onto the environment. The reflection of the known pattern is captured by a receiver and, together with known geometric values, the system can use simple triangulation to establish range measurements. If the receiver measures the position of the reflection along a single axis, we call the sensor an optical triangulation sensor in 1D. If the receiver measures the position of the reflection along two orthogonal axes, we call the sensor a structured light sensor.

Optical Triangulation (1D Sensor)

The principle of optical triangulation in 1D is straightforward, as depicted in **Fig. 1.10**. A collimated

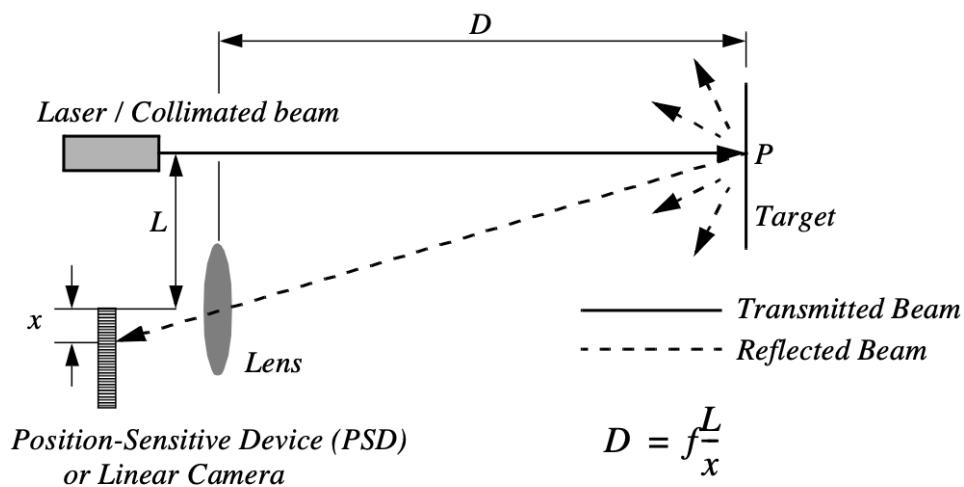


Figure 1.10: Principle of 1D laser triangulation.

beam is transmitted toward the target. The reflected light is collected by a lens and projected onto a position sensitive device¹³ or linear camera. Given the geometry of **Fig. 1.10** the distance D is given by:

$$D = f \frac{L}{x}$$

The distance is proportional to $\frac{1}{x}$, therefore the sensor resolution is best for close objects and becomes worse as distance increases. Sensors based on this principle are used in range sensing up to one or two m, but also in high precision industrial measurements with resolutions far below one μm . Optical triangulation devices can provide relatively high accuracy with very good resolution for close objects. However, the operating range of such a device is normally fairly limited by **geometry**. For



¹³A position sensitive device and/or position sensitive detector is an optical position sensor which can measure a position of a light spot in one or two-dimensions on a sensor surface.

example, an off-the-shelf optical triangulation sensor can operate over a distance range of between 8 cm and 80 cm.

It is inexpensive compared to ultrasonic and laser rangefinder sensors.

Although more limited in range than sonar, the optical triangulation sensor has high bandwidth and does not suffer from cross-sensitivities that are more common in the sound domain.

Structured Light (2D Sensor)

If one replaced the linear camera or Position Sensing Device (PSD) of an optical triangulation sensor with a two-dimensional receiver such as a CCD or CMOS camera, then one can recover distance to a large set of points instead of to only one point. The emitter must project a known pattern, or structured light, onto the environment. Many systems exist which either project light textures, which can be seen in **Fig. 1.12**, or emit collimated light by means of a rotating mirror. Yet another popular alternative is to project a laser stripe by turning a laser beam into a plane using a prism. Regardless of how it is created, the projected light has a known structure, and therefore the image taken by the CCD or CMOS receiver can be filtered to identify the pattern's reflection.

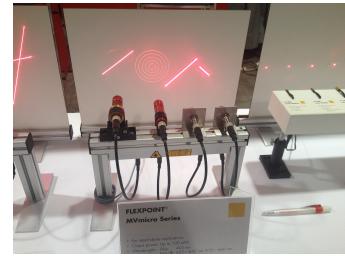


Figure 1.11: Structured light sources on display at the 2014 Machine Vision Show in Boston [5].

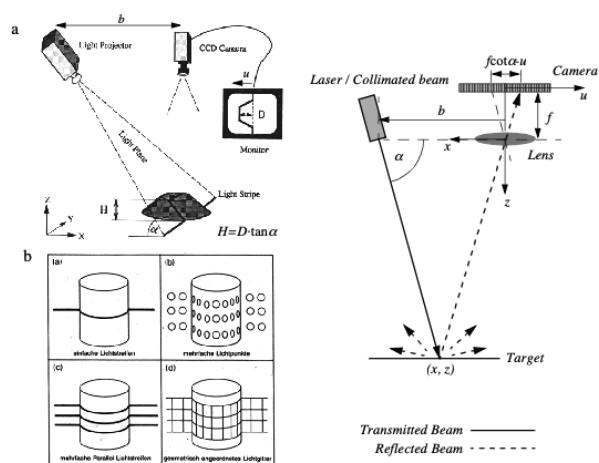


Figure 1.12: a) Principle of active two dimensional triangulation b) Other possible light structures c) One-dimensional schematic of the principle

The problem of recovering depth here far simpler than the problem of passive image analysis.

In passive image analysis, as we discuss later, existing features in the environment must be used to perform correlation, while the present method projects a **known pattern upon the environment** and thereby avoids the standard correlation problem altogether. Furthermore, the structured light sensor

is an active device; so, it will continue to work in dark environments as well as environments in which the objects are featureless¹⁴. In contrast, stereo vision would fail in such texture-free circumstances.

Figure 4.15c shows a one-dimensional active triangulation geometry. We can examine the trade-off in the design of triangulation systems by examining the geometry in figure 4.15c. The measured values in the system are α and u , the distance of the illuminated point from the origin in the imaging sensor.¹⁵ From figure 4.15c, simple geometry shows that:

$$x = \frac{bu}{f \cot \alpha - u} \quad \text{and} \quad z = \frac{bf}{f \cot \alpha - u}.$$

where f is the distance of the lens to the imaging plane. In the limit, the ratio of image resolution to range resolution is defined as the triangulation gain G_p and from equation 4.12 is given by:

$$\frac{\partial u}{\partial z} = G_p = \frac{bf}{z^2}$$

This shows that the ranging accuracy, for a given image resolution, is proportional to source/detector separation b and focal length f , and decreases with the square of the range z . In a scanning ranging system, there is an additional effect on the ranging accuracy, caused by the measurement of the projection angle α . From equation 4.12 we see that:

$$\frac{\partial \alpha}{\partial z} = G_{ff} = \frac{b \sin \alpha^2}{z^2}$$

We can summarise the effects of the parameters on the sensor accuracy as follows:

Baseline Length (b) the smaller b is the more compact the sensor can be. The larger b is the better the range resolution will be. Note also that although these sensors do not suffer from the correspondence problem, the disparity problem still occurs. As the baseline length b is increased, one introduces the chance that, for close objects, the illuminated point(s) may not be in the receiver's field of view.

Detector length and focal length f A larger detector length can provide either a larger field of view or an improved range resolution or partial benefits for both. Increasing the detector length however means a larger sensor head and worse electrical characteristics (increase in random error and reduction of bandwidth). Also, a short focal length gives a large field of view at the expense of accuracy and vice versa.

At one time, laser stripe-based structured light sensors were common on several mobile robot bases as an inexpensive alternative to laser range-finding devices. However, with the increasing quality of laser range-finding sensors in the 1990's the structured light system has become relegated largely to vision research rather than applied mobile robotics.

1.2.2 Motion and Speed Sensors

Some sensors directly measure the relative motion between the robot and its environment. Since such motion sensors detect **relative motion**, so long as an object is moving relative to the robot's

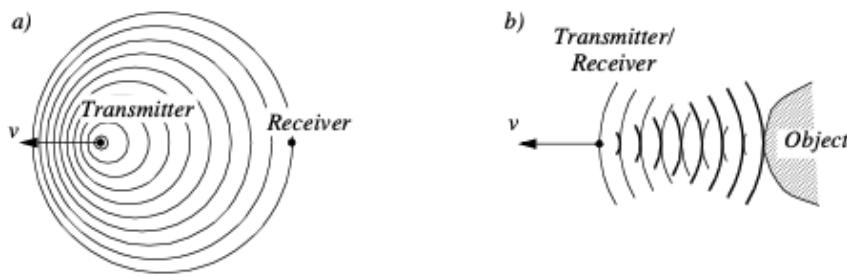
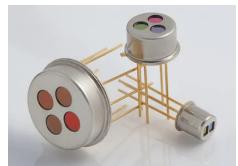


Figure 1.13: Doppler effect between two moving objects (a) or a moving and a stationary object(b)

reference frame, it will be detected and its speed can be estimated. There are a number of sensors that inherently measure some aspect of motion or change.

For example, a pyroelectric¹⁶ sensor detects change in heat.

When someone walks across the sensor's field of view, his motion triggers a change in heat in the sensor's reference frame. In the next subsection, we describe an important type of motion detector based on the **Doppler effect**. These sensors represent a well-known technology with decades of general applications behind them.



¹⁶An example of a pyroelectric sensor.

For fast-moving AMRs such as autonomous highway vehicles and unmanned flying vehicles, Doppler-based motion detectors are the obstacle detection sensor of choice.

Doppler Effect

Anyone who has noticed the change in siren pitch when an ambulance approaches and then passes by is familiar with the Doppler effect.¹⁷

A transmitter emits an electromagnetic or sound wave with a frequency f_t . It is either received by a receiver **Fig. 1.13(a)** or reflected from an object **Fig. 1.13 (b)**. The measured frequency f_r at the receiver is a function of the relative speed v between transmitter and receiver according to

$$f_r = f_t \frac{1}{1 + \frac{v}{c}}$$

if the transmitter is moving and

$$f_r = f_t \left(1 + \frac{v}{c} \right)$$

if the receiver is moving. In the case of a reflected wave **Fig. 1.13 (b)** there is a factor of two introduced, since any change x in relative separation affects the round-trip path length by $2x$.

In such situations it is generally more convenient to consider the change in frequency Δf , known as the Doppler shift, as opposed to the Doppler frequency notation above.

¹⁷For anyone who needs a bit more information, it is the change in the frequency of a wave in relation to an observer who is moving relative to the source of the wave. The Doppler effect is named after the physicist Christian Doppler, who described the phenomenon in 1842. A common example of Doppler shift is the change of pitch heard when a vehicle sounding a horn approaches and recedes from an observer. Compared to the emitted frequency, the received frequency is higher during the approach, identical at the instant of passing by, and lower during the recession.

$$\Delta f = f_t - f_r = \frac{2f_t v \cos \theta}{c} \quad \text{and} \quad v = \frac{\Delta f c}{2f_t \cos \theta}$$

A current application area is both autonomous and manned highway vehicles. Both micro-wave and laser radar systems have been designed for this environment. Both systems have equivalent range, but laser can suffer when visual signals are deteriorated by environmental conditions such as rain, fog, etc. Commercial microwave radar systems are already available for installation on highway trucks. These systems are called VORAD (vehicle on-board radar) and have a total range of approximately 150m. With an accuracy of approximately 97%, these systems report range rate from 0 to 160 km/hr with a resolution of 1 km/ hr. The beam is approximately 4° wide and 5° in elevation. One of the key limitations of radar technology is its bandwidth. Existing systems can provide information on multiple targets at approximately 2 Hz.

1.3 Vision Based Sensors

Vision is our most powerful sense. It provides us with an enormous amount of information about the environment and enables rich, intelligent interaction in dynamic environments. It is therefore not at all surprising that a great deal of effort has been devoted to providing machines with sensors which can at least try to mimic the capabilities of the human vision system.

The first step in this process is the creation of sensing devices that capture the same raw information which is the light the human vision system uses. The main topics which will be described are the two (2) current technologies for creating vision sensors:

1. CCD,
2. CMOS.

Of course, these sensors have specific limitations in performance compared to the human eye, and it is important to understand these limitations. Later sections describe vision-based sensors which are commercially available, similar to the sensors discussed previously, along with their disadvantages and most popular applications.

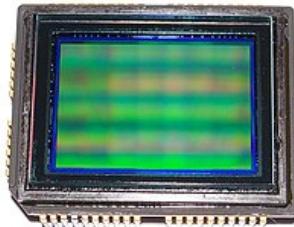


Figure 1.14: Sony ICX493AQ 10.14-megapixel APS-C (23.4 × 15.6 mm) CCD from digital camera Sony DSLR-A200 or DSLR-A300, sensor side [6].

CCD and CMOS Sensors

When it comes to the marketplace, CCD is the most popular fundamental ingredient for robotic vision systems.¹⁸ The CCD chip, which you can see in **Fig. 1.14** is an array of light-sensitive picture elements, or pixels, usually with between 20 000 and 2 million pixels total.

Each pixel can be thought of as a **light-sensitive, discharging capacitor** that is 5 to 25 μm in size. First, the capacitors of all pixels are fully charged, then the integration period begins. As photons of light strike each pixel, the electrons are liberated, which are captured by electric fields and retained at the pixel. Over time, each pixel accumulates a varying level of charge based on the total number of photons that have struck it. After the integration period is complete, the relative charges of all pixels need to be **frozen and read**.

In a CCD, the reading process is performed at one corner of the CCD chip.¹⁹ The bottom row of pixel charges are transported to this corner and read, then the rows above shift down and the process repeats. This means that each charge **must be transported across the chip**, and it is critical the value be preserved.

This requires specialised control circuitry and custom fabrication techniques to ensure the stability of transported charges.

¹⁸Willard Boyle and George E. Smith invented the CCD in 1969 at AT&T Bell Labs. Their original idea was to create a memory device. However, with its publication in 1970, other scientists began experimenting with the technology on a range of applications. Astronomers discovered that they could produce high-resolution images of distant objects, because CCDs offered a photo-sensitivity one hundred times greater than film [7].

¹⁹Because the entire array is read through a single amplifier the output can be highly optimised to give very low noise and extremely high dynamic range. CCDs can have over 100 dB dynamic range with less than 2e of noise [7].

²⁰This also includes CMOS as well.

The photo-diodes used in CCD chips²⁰ are **NOT** equally sensitive to all frequencies of light. They are sensitive to light between 400 nm and 1000 nm wavelength.²¹

²¹This number range is usually given for easier numbers as both CCD and CMOS have sensitivity values at approximately 350 - 1050 nm.

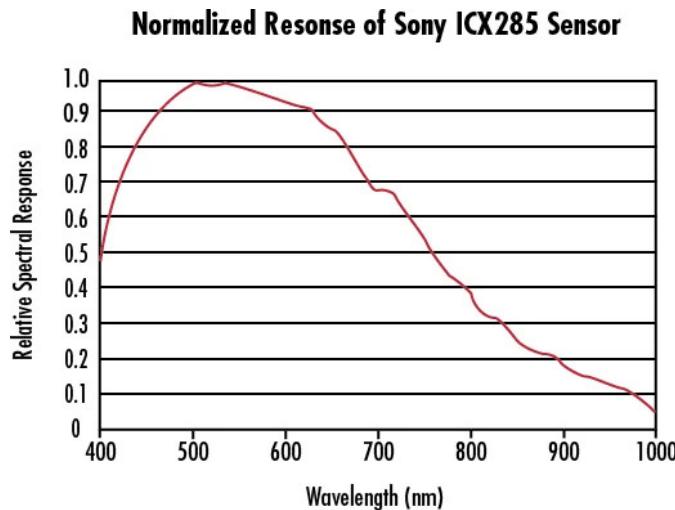


Figure 1.15: Normalized Spectral Response of a Typical Monochrome CCD.

It is important to remember that photodiodes are **less sensitive to the ultraviolet** part of the spectrum and are overly **sensitive to the infrared** portion (e.g. heat) which you can see in Fig. 1.15. You can see that the basic light-measuring process is colourless.²²

There are two (2) common approaches for creating color images. If the pixels on the CCD chip are grouped into 2-by-2 sets of four (4), then red, green and blue dyes can be applied to a colour filter so each individual pixel receives only light of just one color.

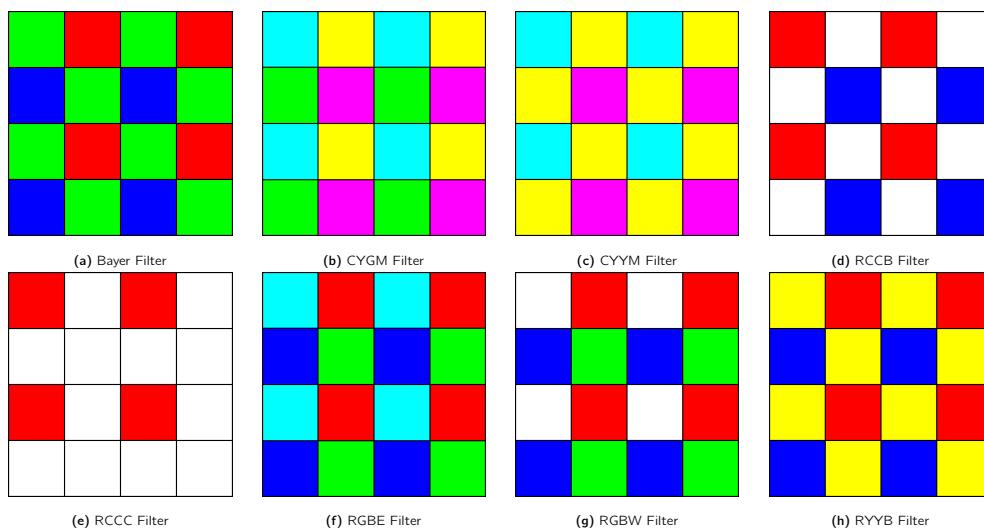


Figure 1.16: Types of colour filter used in commercial and industrial applications

Normally, two (2) pixels measure green while one pixel each measures red and blue light intensity. Of course, this 1-chip color CCD has a geometric resolution disadvantage.

The number of pixels in the system has been effectively cut by a factor of 4, and therefore the image resolution output by the CCD camera will be sacrificed.

The 3-chip color camera avoids these problems by splitting the incoming light into three (3) complete²³ copies. Three separate CCD chips receive the light, with one red, green or blue filter over each entire chip. Thus, in parallel, each chip measures light intensity for just one color, and the camera must combine the CCD chips' outputs to create a joint color image.

²³Albeit, with lower resolution.

Resolution is preserved in this solution, although the 3-chip color cameras are, as one would expect, significantly more expensive and therefore more rarely used in mobile robotics.

Both 3-chip and single chip color CCD cameras suffer from the fact that photo-diodes are much more sensitive to the near-infrared end of the spectrum. This means that the overall system detects blue light much more poorly than red and green. To compensate, the gain must be increased on the blue channel, and this introduces greater absolute noise on blue²⁴ than on red and green. It is not uncommon to assume at least 1 - 2 bits of additional noise on the blue channel.

²⁴This is generally defined as the amplifier noise.

The CCD camera has several camera parameters that affect its behavior. In some cameras, these parameter values are fixed. In others, the values are constantly changing based on built-in feedback loops. In higher-end cameras, the user can modify the values of these parameters via software embedded into the device. The iris position and shutter speed²⁵ regulate the amount of light being measured by the camera. The iris is simply a mechanical aperture that constricts incoming light, just as in standard 35mm cameras. Shutter speed regulates the integration period of the chip. In higher-end cameras, the effective shutter speed can be as brief at 1/30,000s and as long as 2s. Camera gain controls the overall amplification of the analog signal, prior to A/D conversion. However, it is very important to understand that, even though the image may appear brighter after setting high gain, the shutter speed and iris may not have changed at all. Thus gain merely amplifies the signal, and amplifies along with the signal all of the associated noise and error. Although useful in applications where imaging is done for human consumption (e.g. photography, television), gain is of little value to a mobile roboticist.

²⁵It's the speed at which the shutter of the camera closes. A fast shutter speed creates a shorter exposure - the amount of light the camera takes in - and a slow shutter speed gives a longer exposure.

In colour cameras, an additional control exists for white balance. Depending on the source of illumination in a scene²⁶ the relative measurements of red, green and blue light which combine to define pure white light will change dramatically which can be seen in **Fig. 1.17** which can also be adjusted with algorithms [8]. The human eyes compensate for all such effects in ways that are not fully understood, however, the camera can demonstrate glaring inconsistencies in which the same table looks blue in one image, taken during the night, and yellow in another image, taken during the day. White balance controls enable the user to change the relative gain for red, green and blue in order to maintain more consistent color definitions in varying contexts.

²⁶For example this could be fluorescent lamps, incandescent lamps, sunlight, underwater filtered light, etc.

The key disadvantages of CCD cameras are primarily in the areas of inconstancy and **dynamic range**.



Figure 1.17: Example of white balance. Here the same scene is emulated to be shot under different light conditions [9].

Information: Dynamic Range

Dynamic range in photography describes the ratio between the maximum and minimum measurable light intensities (white and black, respectively). In the real world, one never encounters true white or black - only varying degrees of light source intensity and subject reflectivity. Therefore the concept of dynamic range becomes more complicated, and depends on whether you are describing a capture device (such as a camera or scanner), a display device (such as a print or computer display), or the subject itself.

As mentioned above, a number of parameters can change the brightness and colours with which a camera creates its image.

Manipulating these parameters in a way to provide consistency over time and over environments, for example ensuring a green shirt always looks green, and something dark grey is always dark grey, remains an open problem [10].

The second type of disadvantages relates to the behavior of a CCD chip in environments with **extreme illumination**. In cases of very low illumination, each pixel will receive only a small number of photons. The longest possible shutter speed and camera optics (i.e. pixel size, chip size, lens focal length and diameter) will determine the minimum level of light for which the signal is stronger than random error noise. In cases of very high illumination, a pixel fills its well with free electrons and, as the well reaches its limit, the probability of trapping additional electrons falls and therefore the linearity between incoming light and electrons in the well degrades. This is termed saturation²⁷ and can indicate the existence of a further problem related to cross-sensitivity [12]. When a well has reached its limit, then additional light within the remainder of the integration period may cause further charge to leak into neighbouring pixels, causing them to report incorrect values or even reach secondary saturation. This effect, called blooming, means that individual pixel values are **NOT** truly **independent**. The camera parameters may be adjusted for an environment with a particular light level, but the problem remains that the dynamic range of a camera is limited by the well capacity of the individual pixels.



²⁷Example of blooming caused by saturation of a sensor pixel. The sun is so bright in the image that there is blooming on the sun itself, leaking into the surrounding pixels, and a vertical smear across the whole image [11].

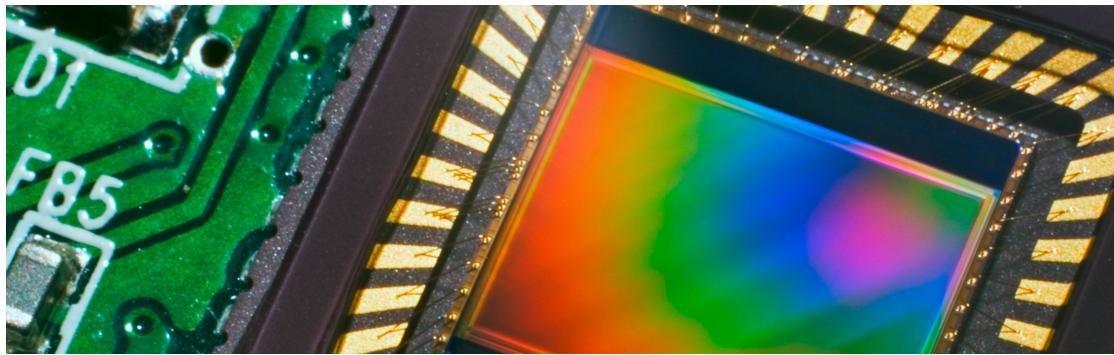


Figure 1.18: A close-up view of a CMOS sensor and its circuitry [13].

For example, a high quality CCD may have pixels that can hold 40 000 electrons. The noise level for reading the well may be 11 electrons, and therefore the dynamic range will be 40,000:11, or 3,600:1, which is 35 dB.

1.3.1 CMOS Technology

The Complementary Metal Oxide Semiconductor (CMOS) chip is a significant departure from the CCD. Similar to CCD, it too has an array of pixels, but located alongside each pixel are **several transistors specific to that pixel**. Just as in CCD chips, all of the pixels accumulate charge during the integration period. During the data collection step, the CMOS takes a new approach:

The pixel-specific circuitry next to every pixel measures and amplifies the pixel's signal, all in parallel for every pixel in the array.

Using more traditional traces from general semiconductor chips, the resulting pixel values are all carried to their destinations. CMOS has a number of advantages over CCD technologies. First and foremost, there is no need for the specialized clock drivers and circuitry required in the CCD to transfer each pixel's clock down all of the array columns and across all of its rows.²⁸

This also means that specialized semiconductor manufacturing processes are not required to create CMOS chips.

Therefore, the same production lines that create microchips can create inexpensive CMOS chips as well. The CMOS chip is so much simpler that it consumes significantly less power, it operates with a power consumption a tenth the power consumption of a CCD chip [15].

In a AMR, power is a scarce resource and therefore this is an important advantage.

On the other hand, the CMOS chip also faces several disadvantages.

- Most importantly, the circuitry next to each pixel consumes valuable real estate on the face of the light-detecting array. Many photons hit the transistors rather than the photodiode, making



²⁸-CAM80CUNX is an 8MP Ultra-lowlight MIPI CSI-2 camera capable of streaming 4K @ 44 fps. This 8MP camera is based on SONY STARVIS IMX415 CMOS image sensor [14]

the CMOS chip significantly less sensitive than an equivalent CCD chip.

- CMOS, compared to CCD is still finding ground in the marketplace, and as a result, the best resolution that one can purchase in CMOS format continues to be far inferior to the best CCD chips available.
- CMOS sensors have a lower dynamic range,
- CMOS sensors have higher levels of noise.

Compared to the human eye, these chips all have worse performance, cross-sensitivity and a limited dynamic range. As a result, vision sensors today continue to be fragile. Only over time, as the underlying performance of imaging chips improves, will significantly more robust vision-based sensors for AMRs be available.

Information: Shot Noise

Shot noise or Poisson noise is a type of noise which can be modeled by a Poisson process. In electronics shot noise originates from the discrete nature of electric charge. Shot noise also occurs in photon counting in optical devices, where shot noise is associated with the particle nature of light.

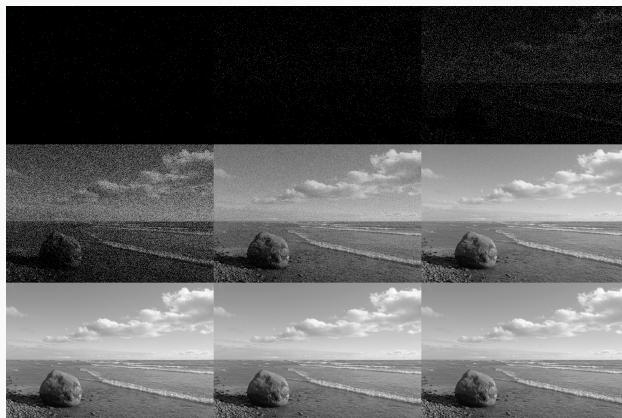


Figure 1.19: Photon noise simulation. Number of photons per pixel increases from left to right and from upper row to bottom row [16].

1.3.2 Visual Ranging Sensors

Range sensing is extremely important in AMR as it is a basic input for successful obstacle avoidance. As we have seen earlier, a number of sensors are popular in robotics specifically for their ability to recover depth estimates:

ultrasonic, laser rangefinder, optical rangefinder, etc.

It is natural to attempt to implement ranging functionality using vision chips as well. However, a fundamental problem with visual images makes rangefinding relatively difficult.

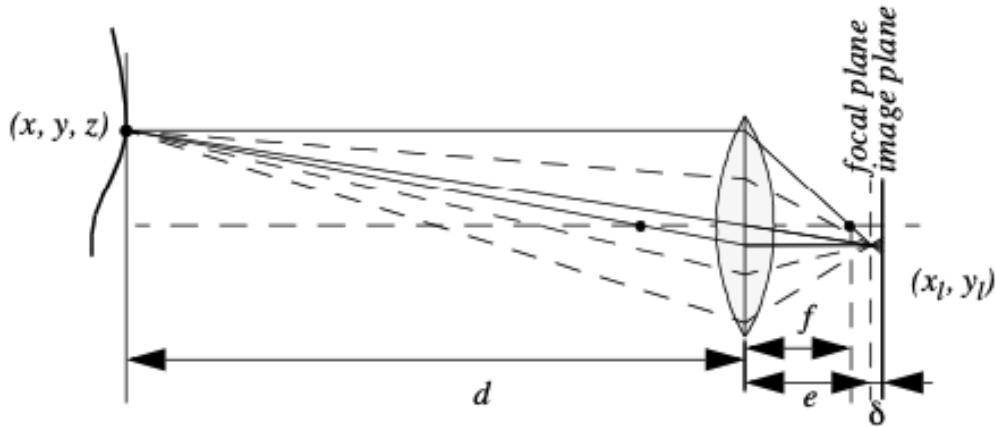


Figure 1.20: Depiction of the camera optics and its impact on the image. To get a sharp image, the image plane must coincide with the focal plane. Otherwise the image of the point (x, y, z) will be blurred in the image as can be seen in the drawing above.

Any vision chip collapses the three-dimensional world into a two-dimensional image plane, thereby losing depth information. If one can make strong assumptions regarding the size of objects in the world, or their particular colour and reflectance, then one can directly interpret the appearance of the two-dimensional image to recover depth. But such assumptions are rarely possible in real-world AMR applications.

Without such assumptions, a single picture does not provide enough information to recover spatial information.

The general solution is to recover depth by looking at several images of the scene to gain more information, which will be hopefully enough to at least partially recover depth. The images used **must be different**, so that taken together they provide additional information. They could differ in viewpoint, which would allow the use of stereo or motion algorithms.

An alternative is to create different images, not by changing the viewpoint, but by changing the camera geometry, such as the focus position or lens iris. This is the fundamental idea behind depth from focus and depth from defocus techniques. We will now look into the general approach to the depth from focus techniques as it presents a straightforward and efficient way to create a vision-based range sensor.

1.3.3 Depth from Focus

The depth from focus class of techniques relies on the fact that image properties not only change as a function of the **scene**, but also as a function of the **camera parameters**. The relationship between camera parameters and image properties is depicted in **Fig. 1.20**. The fundamental formula governing image formation relates the distance of the object from the lens, **d** in **Fig. 1.20**, to the

distance e from the lens to the focal point, based on the focal length f of the lens:

$$\frac{1}{f} = \frac{1}{d} + \frac{1}{e}$$

²⁹A three-dimensional counterpart to a pixel. If the image plane is located at distance e from the lens, then for the specific object voxel²⁹ depicted, all light will be focused at a single point on the image plane and the object voxel will be focused. However, when the image plane is **NOT** at e , as is seen in **Fig. 1.20**, then the light from the object voxel will be cast on the image plane as a **blur circle**. To a first approximation, the light is homogeneously distributed throughout this blur circle, and the radius R of the circle can be characterized according to the equation:

$$R = \frac{L\delta}{2e}$$

where L is the diameter of the lens or aperture and δ is the displacement of the image plan from the focal point.

Given these formulae, several basic optical effects are clear.

³⁰The aperture is the opening in the lens that allows light to enter the camera and onto the sensor or film.

For example, if the aperture³⁰ or lens is reduced to a point, as in a pin-hole camera, then the radius of the blur circle approaches zero.

This is consistent with the fact that decreasing the iris aperture opening causes the depth of field to increase until all objects are in focus. Of course, the disadvantage of doing so is that we are allowing less light to form the image on the image plane and so this is practical only in bright circumstances. The second property to be deduced from these optics equations relates to the sensitivity of blurring as a function of the distance from the lens to the object.

Suppose the image plane is at a fixed distance 1.2 from a lens with diameter $L = 0.2$ and focal length $f = 0.5$. We can see from Equation (4.20) that the size of the blur circle R changes proportionally with the image plane displacement . If the object is at distance $d = 1$, then from Equation (4.19) we can compute $e=1$ and therefore $\delta = 0.2$. Increase the object distance to $d = 2$ and as a result $\delta = 0.533$. Using Equation (4.20) in each case we can compute $R = 0.02$ $R = 0.08$ respectively. This demonstrates high sensitivity for defocusing when the object is close to the lens. In contrast suppose the object is at $d = 10$. In this case we compute $e = 0.526$. But if the object is again moved one unit, to $d = 11$, then we compute $e = 0.524$. Then resulting blur circles are $R = 0.117$ and $R =$



Figure 1.21: Three images of the same scene taken with a camera at three different focusing positions. Note the significant change in texture sharpness between the near surface and far surface [17].

0.129, far less than the quadrupling in R when the obstacle is 1/10 the distance from the lens. This analysis demonstrates the fundamental limitation of depth from focus techniques: they lose sensitivity as objects move further away (given a fixed focal length). Interestingly, this limitation will turn out to apply to virtually all visual ranging techniques, including depth from stereo and depth from motion. Nevertheless, camera optics can be customised for the depth range of the intended application. For example, a "zoom" lens with a very large focal length f will enable range resolution at significant distances, of course at the expense of field of view. Similarly, a large lens diameter, coupled with a very fast shutter speed, will lead to larger, more detectable blur circles. Given the physical effects summarised by the above equations, one can imagine a visual ranging sensor that makes use of multiple images in which camera optics are varied (e.g. image plane displacement) and the same scene is captured (see Fig. 4.20). In fact this approach is not a new invention. The human visual system uses an abundance of cues and techniques, and one system demonstrated in humans is depth from focus. Humans vary the focal length of their lens continuously at a rate of about 2 Hz. Such approaches, in which the lens optics are actively searched in order to maximise focus, are technically called depth from focus. In contrast, depth from defocus means that depth is recovered using a series of images that have been taken with different camera geometries. Depth from focus methods are one of the simplest visual ranging techniques. To determine the range to an object, the sensor simply moves the image plane (via focusing) until maximizing the sharpness of the object. When the sharpness is maximised, the corresponding position of the image plane directly reports range. Some autofocus cameras and virtually all autofocus video cameras use this technique. Of course, a method is required for measuring the sharpness of an image or an object within the image. The most common techniques are approximate measurements of the sub-image gradient:

$$\text{sharpness}_1 = \sum_{x,y} |I(x, y) - I(x-1, y)| \quad (1.3)$$

$$\text{sharpness}_2 = \sum_{x,y} (I(x, y) - I(x-2, y-2))^2 \quad (1.4)$$

A significant advantage of the horizontal sum of differences technique (Equation (4.21)) is that the calculation can be implemented in analog circuitry using just a rectifier, a low-pass filter and a high-pass filter. This is a common approach in commercial cameras and video recorders. Such systems will be sensitive to contrast along one particular axis, although in practical terms this is rarely an issue. However depth from focus is an active search method and will be slow because it takes time to change the focusing parameters of the camera, using for example a servo-controlled focusing ring. For this reason this method has not been applied to AMRs. A variation of the depth from focus technique has been applied to a AMR, demonstrating obstacle avoidance in a variety of environments as well as avoidance of concave obstacles such as steps and ledges [95]. This robot uses three monochrome cameras placed as close together as possible with different, fixed lens focus positions (Fig. 4.21).

Several times each second, all three frame-synchronised cameras simultaneously capture three images of the same scene. The images are each divided into five columns and three rows, or 15 subregions. The approximate sharpness of each region is computed using a variation of Equation (4.22), leading to a total of 45 sharpness values. Note that Equation 22 calculates sharpness along diagonals but skips one row. This is due to a subtle but important issue. Many cameras produce images in

interlaced mode. This means that the odd rows are captured first, then afterwards the even rows are captured. When such a camera is used in dynamic environments, for example on a moving robot, then adjacent rows show the dynamic scene at two different time points, differing by up to 1/30 seconds. The result is an artificial blurring due to motion and not optical defocus. By comparing only even-number rows we avoid this interlacing side effect.

Recall that the three images are each taken with a camera using a different focus position. Based on the focusing position, we call each image close, medium or far. A 5x3 coarse depth map of the scene is constructed quickly by simply comparing the sharpness values of each three corresponding regions. Thus, the depth map assigns only two bits of depth information to each region using the values close, medium and far. The critical step is to adjust the focus positions of all three cameras so that flat ground in front of the obstacle results in medium readings in one row of the depth map. Then, unexpected readings of either close or far will indicate convex and concave obstacles respectively, enabling basic obstacle avoidance in the vicinity of objects on the ground as well as drop-offs into the ground. Although sufficient for obstacle avoidance, the above depth from focus algorithm presents unsatisfyingly coarse range information. The alternative is depth from defocus, the most desirable of the focus-based vision techniques. Depth from defocus methods take as input two or more images of the same scene, taken with different, known camera geometry. Given the images and the camera geometry settings, the goal is to recover the depth information of the three-dimensional scene represented by the images. We begin by deriving the relationship between the actual scene properties (irradiance and depth), camera geometry settings and the image g that is formed at the image plane. The focused image $f(x,y)$ of a scene is defined as follows. Consider a pinhole aperture ($L=0$) in lieu of the lens. For every point p at position (x,y) on the image plane, draw a line through the pinhole aperture to the corresponding, visible point P in the actual scene. We define $f(x,y)$ as the irradiance (or light intensity) at p due to the light from P . Intuitively, $f(x,y)$ represents the intensity image of the scene perfectly in focus

1.4 Feature Extraction

An AMR must be able to determine its relationship to the environment by making measurements with its sensors and then using those measured signals. A wide variety of sensing technologies are available, as we discussed previously. But every sensor we have presented is imperfect:

measurements always have error and, therefore, uncertainty associated with them.

Therefore, sensor inputs must be used in a way that enables the robot to interact with its environment successfully in spite of measurement uncertainty. There are two (2) strategies for using uncertain sensor input to guide the robot's behavior. One strategy is to use each sensor measurement as a raw and individual value. Such raw sensor values could for example be tied directly to robot behavior, whereby the robot's actions are a function of its sensor inputs. Alternatively, the raw sensors values could be used to update an intermediate model, with the robot's actions being triggered as a function of this model rather than the individual sensor measurements.

The second strategy is to extract information from one or more sensor readings first, generating a higher-level percept that can then be used to inform the robot's model and perhaps the robot's actions directly. We call this process feature extraction, and it is this next, optional step in the perceptual interpretation pipeline (Fig. 4.34) that we will now discuss.

In practical terms, mobile robots do not necessarily use feature extraction and scene interpretation for every activity. Instead, robots will interpret sensors to varying degrees depending on each specific functionality. For example, in order to guarantee emergency stops in the face of immediate obstacles, the robot may make direct use of raw forward-facing range readings to stop its drive motors. For local obstacle avoidance, raw ranging sensor strikes may be combined in an occupancy grid model, enabling smooth avoidance of obstacles meters away. For map-building and precise navigation, the range sensor values and even vision sensor measurements may pass through the complete perceptual pipeline, being subjected to feature extraction followed by scene interpretation to minimize the impact of individual sensor uncertainty on the robustness of the robot's map-making and navigation skills. The pattern that thus emerges is that, as one moves into more sophisticated, long-term perceptual tasks, the feature extraction and scene interpretation aspects of the perceptual pipeline become essential.

1.4.1 Defining Feature

Features are recognizable structures of elements in the environment. They usually can be extracted from measurements and mathematically described. Good features are always perceivable and easily detectable from the environment. We distinguish between low-level features (geometric primitives) like lines, circles or polygons and high-level features (objects) such as edges, doors, tables or a trash can. At one extreme, raw sensor data provides a large volume of data, but with low distinctiveness of each individual quantum of data. Making use of raw data has the potential advantage that every bit of information is fully used, and thus there is a high conservation of information. Low level

features are abstractions of raw data, and as such provide a lower volume of data while increasing the distinctiveness of each feature. The hope, when one incorporates low level features, is that the features are filtering out poor or useless data, but of course it is also likely that some valid information will be lost as a result of the feature extraction process. High level features provide maximum abstraction from the raw data, thereby reducing the volume of data as much as possible while providing highly distinctive resulting features. Once again, the abstraction process has the risk of filtering away important information, potentially lowering data utilization.

Although features must have some spatial locality, their geometric extent can range widely. For example, a corner feature inhabits a specific coordinate location in the geometric world. In contrast, a visual "fingerprint" identifying a specific room in an office building applies to the entire room, but has a location that is spatially limited to the one, particular room. In mobile robotics, features play an especially important role in the creation of environmental models. They enable more compact and robust descriptions of the environment, helping a mobile robot during both map-building and localization. When designing a mobile robot, a critical decision revolves around choosing the appropriate features for the robot to use. A number of factors are essential to this decision:

Target Environment For geometric features to be useful, the target geometries must be readily detected in the actual environment. For example, line features are extremely useful in office building environments due to the abundance of straight walls segments while the same feature is virtually useless when navigating Mars.

Available Sensors Obviously the specific sensors and sensor uncertainty of the robot impacts the appropriateness of various features. Armed with a laser rangefinder, a robot is well qualified to use geometrically detailed features such as corner features due to the high quality angular and depth resolution of the laser scanner. In contrast, a sonar-equipped robot may not have the appropriate tools for corner feature extraction.

Computational Power Vision-based feature extraction can effect a significant computational cost, particularly in robots where the vision sensor processing is performed by one of the robot's main processors.

Environment representation Feature extraction is an important step toward scene interpretation, and by this token the features extracted must provide information that is consonant with the representation used for the environment model. For example, non-geometric vision-based features are of little value in purely geometric environment models but can be of great value in topological models of the environment. Figure 4.35 shows the application of two different representations to the task of modeling an office building hallway. Each approach has advantages and disadvantages, but extraction of line and corner features has much more relevance to the representation on the left. Refer to Chapter 5, Section 5.5 for a close look at map representations and their relative tradeoffs. In the following two sections, we present specific feature extraction techniques based on the two most popular sensing modalities of mobile robotics: range sensing and visual appearance-based sensing.

1.4.2 Using Range Data

Most of today's features extracted from ranging sensors are geometric primitives such as line segments or circles. The main reason for this is that for most other geometric primitives the parametric description of the features becomes too complex and no closed form solution exists. Here we will describe line extraction in detail, demonstrating how the uncertainty models presented above can be applied to the problem of combining multiple sensor measurements. Afterwards, we briefly present another very successful feature for indoor mobile robots, the corner feature, and demonstrate how these features can be combined in a single representation.

Line Extraction

Geometric feature extraction is usually the process of comparing and matching measured sensor data against a predefined description, or template, of the expected feature. Usually, the system is overdetermined in that the number of sensor measurements exceeds the number of feature parameters to be estimated. Since the sensor measurements all have some error, there is no perfectly consistent solution and, instead, the problem is one of optimization. One can, for example, extract the feature that minimizes the discrepancy with all sensor measurements used (e.g. least squares estimation). In this section we present an optimization-based solution to the problem of extracting a line feature from a set of uncertain sensor measurements. For greater detail than is presented below, refer to [19], pp. 15 and 221.

Probabilistic Line Extraction

4.36. There is uncertainty associated with each of the noisy range sensor measurements, and so there is no single line that passes through the set. Instead, we wish to select the best possible match, given some optimization criterion. More formally, suppose n ranging measurement points in polar coordinates $x = (\rho, \theta)$ are produced by the robot's sensors. We know that there is uncertainty associated with each measurement, and so we can model each measurement using two random variables $X = (P, Q)$. In this analysis we assume that uncertainty with respect to the actual value θ of P and Q are independent. Based on Equation (4.56) we can state this formally: Furthermore, we will assume that each random variable is subject to a Gaussian probability density curve, with a mean at the true value and with some specified variance: Given some measurement point (ρ, θ) , we can calculate the corresponding Euclidean coordinates $x = (\cos \theta, \sin \theta)$. If there were no error, we would want to find a line for which all measurements lie on that line: Of course there is measurement error, and so this quantity will not be zero. When it is non-zero, this is a measure of the error between the measurement point (ρ, θ) and the line, specifically in terms of the minimum orthogonal distance between the point and the line. It is always important to understand how the error that shall be minimized is being measured. For example a number of line extraction techniques do not minimize this orthogonal point-line distance, but instead the distance parallel to the y -axis between the point and the line. A good illustration of the variety of

optimization criteria is available in [18] where several algorithms for fitting circles and ellipses are presented which minimize algebraic and geo-metric distances. For each specific (x_i, y_i) , we can write the orthogonal distance d between (x_i, y_i) and the line as:

Chapter 2

Mobile Robot Localisation

Table of Contents

2.1	Introduction	41
2.2	The problems of Noise and Aliasing	43
2.3	Localisation v. Hard-Coded Navigation	48
2.4	Representing Belief	51
2.5	Representing Maps	55
2.6	Probabilistic Map-Based Localisation	64
2.7	Other Examples of Localisation Methods	77
2.8	Building Maps	82

2.1 Introduction



Figure 2.1: Navigation is one if not the most demanding and complicated task in AMR. However a successful implementation will result in a versatile AMR which can find its way in unknown environments such as exploring other planets [18].

Navigation is one of, if not, the most challenging problem faced by an AMR and for the robot to be able to successfully navigate its environment, it requires four (4) functions:

Perception the robot must be able to interpret its sensors to extract meaningful data,

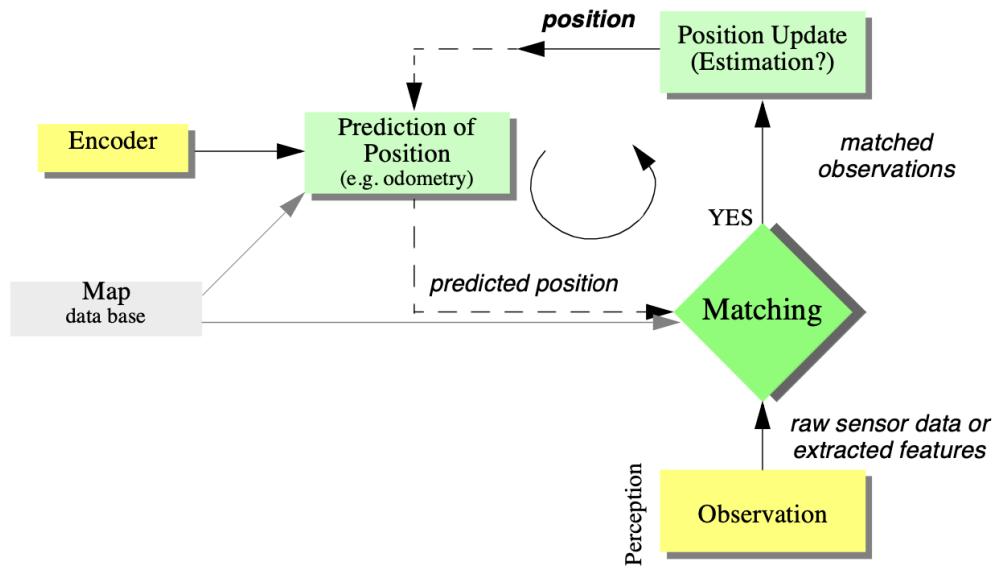


Figure 2.2: General schematic for mobile robot localisation.

Localisation the robot must be able to determine its position within the environment,

Cognition the robot must be able to decide how to act to achieve its goals,

Motion control the robot must be able to modulate its motor outputs to achieve the desired trajectory.

Of these four (4) aforementioned components, localisation has received the greatest research attention in the past and, as a result, significant advances have been made on this front, presented in [19], [20], and [21]. In this chapter, we will explore the successful localisation methodologies and techniques used in academic research and industrial application [22].

The structure of the chapter is as follows:

- We will describe how sensor and effector uncertainty is responsible for the difficulties of localisation in Section 2.2,
- Then, in Section 2.3, we will have a look at the two (2) extreme approaches to dealing with the challenge of robot localisation [23]:
 - Avoiding localisation altogether,
 - Performing explicit map-based localisation
- The remainder of the chapter discusses the question of representation, which we will have a look at different case studies of successful localisation systems using a variety of representations and techniques to achieve AMR localisation.

2.2 The problems of Noise and Aliasing

If one could attach an accurate GPS sensor to an AMR, much of the localisation problem would be obviated. GPS would then inform the robot of its **exact** position and orientation, indoors and outdoors, so the answers to the questions,

Where am I?, Where am I going?, and, How should I get there? [24]

would **always** be immediately available.

Unfortunately, such a sensor is **NOT** currently practical.¹ The existing GPS network provides accuracy to within several m [25], which is still not the optimal accuracy for localising human-scale AMRs as well as miniature AMRs such as desk robots and the body-navigating nano-robots of the future.

In addition, GPS cannot function indoors or in obstructed areas and are therefore limited in their workspace. But, looking beyond the limitations of GPS, localisation implies more than knowing one's absolute position in the Earth's reference frame.

Consider a robot which is interacting with humans. This robot may need to identify its absolute position, but its relative position with respect to target humans is also equally important. Its localisation task can include:

- identifying humans using its sensor array [26],
- then computing its relative position to the humans.

Furthermore, during operation a robot will select a strategy for achieving its goals. If it intends to reach a particular location, then localisation may not be enough. The robot may need to acquire or build an environmental model,² which aids it in planning a path to the goal.

Localisation means more than simply determining an absolute pose in space. It means building a map, then identifying the robot's position relative to that map.

¹Of course, this misleading statement as we have technology which allows the shrinking of errors down to cm using real-time kinematic positioning which is used to correct Global Navigation Satellite System (GNSS), which transmits the robot's location by longitude, latitude, altitude, and a timestamp [21].

²i.e., a map representing 2D space if it is an indoor space which is level, or a 3D space if it is navigating rough terrain.

Clearly, the robot's sensors and effectors play an integral role in all the above forms of localisation. It is because of the inaccuracy and incompleteness of these sensors and effectors localisation poses difficult challenges.

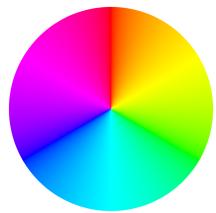
2.2.1 Sensor Noise

Sensors are the fundamental robot input for the process of perception, and therefore the degree to which sensors can discriminate world state is critical. Sensor noise produces a **limitation on the consistency of sensor readings** in the same environmental state and, therefore, on the number of

useful bits available from each sensor reading.

Often, the source of sensor noise problems is that some environmental features are not captured by the robot's representation and are thus overlooked.

³For example, this could be indoor office building, or a warehouse.



⁴One of the properties of a colour, defined as the degree to which a stimulus can be described as similar to or different from stimuli that are described as red, orange, yellow, green, blue, violet within certain theories of colour vision.

For example, a vision system used for indoor navigation³ may use the colour values detected by its colour CCD camera. When the Sun is hidden by clouds, the illumination of the building's interior changes due to windows throughout the building. As a result, hue⁴ values are not constant. The colour CCD appears noisy from the robot's perspective as if subject to **random error**, and the hue values obtained from the CCD camera will be unusable, unless the robot is able to note the position of the Sun and clouds in its representation.

Illumination dependency is only one example of the apparent noise in a vision-based sensor system [27]. Picture jitter, signal gain, blooming and blurring are all additional sources of noise, potentially reducing the useful content of a colour video image.

Consider the noise level of ultrasonic range-measuring sensors, such as sonars, as we discussed previously. When a sonar transducer emits sound towards a relatively smooth and angled surface, much of the signal will coherently reflect away, failing to generate a return echo. Depending on the material characteristics, a small amount of energy may return nonetheless. When this level is close to the gain threshold of the sonar sensor, then the sonar will, at times, succeed and, at other times, fail to detect the object. From the robot's perspective, a virtually unchanged environmental state will result in two (2) different possible sonar readings:

one short, and one long which causes an nondeterministic behaviour.

⁵The propagation phenomenon resulting in signals reaching the receiver by two (2) or more paths. Causes of multipath can be atmospheric ducting, ionospheric reflection and refraction, and reflection from water bodies and terrestrial objects such as mountains and buildings.

⁶Sensor fusion is the process of using information from several different sensors to estimate the state of a dynamic system. The resulting estimate is, in some senses, better than it would be if the sensors were used individually [28].

The poor Signal-to-Noise Ratio (SNR) of a sonar sensor is further confounded by interference between multiple sonar emitters. Often, research robots have between 12 to 48 sonars on a single platform. In acoustically reflective environments, multipath interference⁵ is possible between the sonar emissions of one transducer and the echo detection circuitry of another transducer. The result can be dramatically large errors in ranging values due to a set of coincidental angles. Such errors occur rarely, less than 1% of the time, and are virtually random from the robot's perspective.

In conclusion, sensor noise reduces the useful information content of sensor readings. Clearly, the solution is to take multiple readings into account, employing temporal fusion or multi-sensor fusion⁶ to increase the overall information content of the robot's inputs.

2.2.2 Sensor Aliasing

Aliasing is the second major shortcoming of AMR sensors which cause them to give little information content, further amplifying the problem of **perception** and **localisation**.

Information: The Human Experience

The problem, known as sensor aliasing, is a phenomenon that humans seldom encounter. The human sensory system, particularly the visual system, tends to receive unique inputs in each unique local state within normal usage [29]. In other words, every different place looks different. The power of this unique mapping is only apparent when one considers situations where this fails to hold.

Consider moving through an unfamiliar building that is completely dark. When the visual system sees only black, one's localisation system quickly degrades. Another useful example is that of a human-sized maze made from tall hedges. Such mazes have been created for centuries, and humans find them extremely difficult to solve without landmarks or clues because, without visual uniqueness, human localisation competence degrades rapidly.

In robots, the non-uniqueness of sensors readings, or sensor aliasing⁷, is the norm and not the exception. Consider a narrow-beam rangefinder such as ultrasonic or infrared rangefinders. This sensor provides range information in a single direction without any additional data regarding material composition such as **color**, **texture** and **hardness**. Even for a robot with several such sensors in an array, there are a variety of environmental states that would trigger the same sensor values across the array. Formally, there is a many-to-one mapping from environmental states to the robot's perceptual inputs. Therefore, the robot's sensors cannot distinguish from among these many states.

A classical problem with sonar-based robots involves distinguishing between humans and inanimate objects in an indoor setting [31, 32].

When facing an apparent obstacle in front of itself, should the robot say "Excuse me" because the obstacle may be a moving human, or should the robot plan a path around the object because it may be a cardboard box?

With sonar alone, these states are aliased and differentiation is impossible.



⁷Sensor aliasing in multiple types of sensors. One of the most apparent one is usually seen in digital images. For example, in the image above, due to low sampling, moire pattern starts to be seen [30].

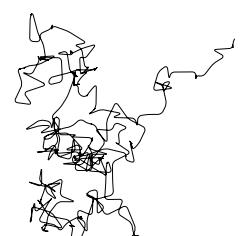
The navigation problem due to sensor aliasing is that, even with noise-free sensors, the amount of information is generally **insufficient** to identify the robot's accurate position from a single sensor's reading. Therefore techniques needs to be employed by the robot programmer which base the robot's localisation on a series of readings and **sufficient information** to recover the robot's position over time.

2.2.3 Effector Noise

The challenges of localisation does **NOT** lie with sensor technologies alone. Just as robot sensors are noisy, limiting the information content of the signal, so do the robot effectors.

A single action taken by a AMR may have several different possible results, even though from the robot's point of view the initial state before the action was taken is well-known.

In short, AMR effectors introduce uncertainty about future state.⁸ The simple act of moving tends to **increase the uncertainty** of a AMR. There are, of course, exceptions. Using filters and predictive modelling, the motion can be carefully planned so as to minimise this effect, and indeed sometimes to actually result in more certainty. Furthermore, when the robot actions are taken in concert with



⁸An over-exaggerated example of effector noise where the motion is severely affected by the uncertainty caused by the deterministic error.

careful interpretation of sensory feedback, it can compensate for the uncertainty introduced by noisy actions using the information provided by the sensors.

First, however, it is important to understand the precise nature of the effector noise that impacts AMR. It is important to note that, from the robot's point of view, this error in motion is viewed as **error in the odometer**, or the robot's inability to estimate its own position over time using knowledge of its kinematics and dynamics. The true source of error generally lies in an **incomplete model** of the environment.

For instance, the robot does **NOT** model the fact that the floor may be sloped, the wheels may slip, and a human may push the robot.

All of these unmodeled sources of error result in:

- inaccuracy between the physical motion of the robot,
- the intended motion of the robot, and the
- proprioceptive sensor estimates of motion.

⁹The process of calculating the current position of a moving object by using a previously determined position, or fix, and incorporating estimates of speed, heading (or direction or course), and elapsed time.

In odometry and dead reckoning⁹ the position update is based on proprioceptive sensors. The movement of the robot, sensed with wheel encoders and /or heading sensors is integrated to compute position. Because the sensor measurement errors are integrated, the position error accumulates over time. Thus the position has to be updated from time to time by other localisation mechanisms. Otherwise the robot is not able to maintain a meaningful position estimate in long run.

In the following we will concentrate on odometry based on the wheel sensor readings of a differential drive robot only [33].¹⁰

There are many sources of odometric error, from environmental factors to resolution:

- Limited resolution during integration¹¹
- Misalignment of wheels causing **deterministic** error,
- Unequal wheel diameter, which again, causing **deterministic** error,
- Unequal floor contact, which can cause **slipping** during operation.

¹²To reiterate, deterministic errors are any errors which can be avoided and are generally caused by bad design or poorly calibrated sensors.

Some of the errors might be deterministic¹² (systematic). However, there are still a number of non-deterministic (random) errors which remain, leading to uncertainties in position estimation over time. From a geometric point of view one can classify the errors into three (3) types:

Range error Integrated path length of the robot movement, as in the sum of wheel motion.

Turn error Similar to range error, but for turns which are difference of the wheel motions.

Drift error difference in the error of the wheels leads to an error in the robot's angular orientation.

Over long periods of time, turn and drift errors far outweigh range errors, as their contribute to the overall position error is non-linear. Consider a robot, whose position is initially perfectly well-known, moving forward in a straight line along the x axis. The error in the y -position introduced by a move of d meters will have a component of $d \sin \Delta\theta$, which can be quite large as the angular error $\Delta\theta$ grows. Over time, as an AMR moves about the environment, the rotational error between its internal reference frame and its original reference frame grows quickly. As the robot moves away from the origin of these reference frames, the resulting linear error in position grows quite large. It is instructive to establish an error model for odometric accuracy and see how the errors propagate over time.

2.3 Localisation v. Hard-Coded Navigation

Fig. 2.3 depicts a standard indoor environment an AMR is set to navigate. Now, suppose an AMR in question must deliver messages between two (2) specific rooms in this environment:

These are rooms A and B.

In creating a navigation system for this task, it is clear the AMR will need sensors and a motion control system. Sensors are required to avoid hitting moving obstacles such as humans, and some motion control system is required so that the robot can actively move.

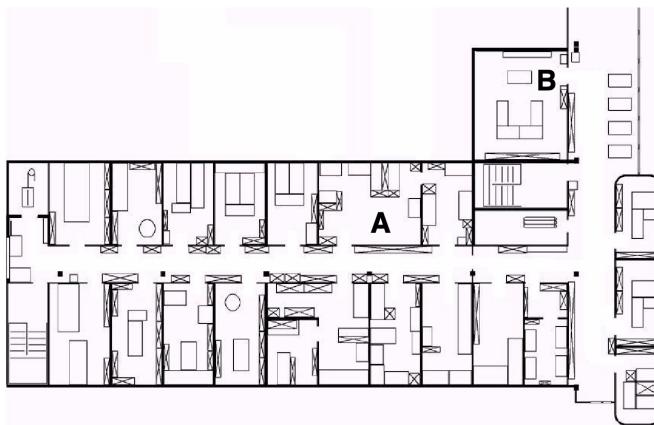


Figure 2.3: A sample environment.

It is less evident, however, whether or not this AMR will require a localisation system. Localisation may seem mandatory to successfully navigate between the two (2) rooms. It is through localising on a map, after all, which the robot can hope to recover its position and detect when it has arrived at the goal location. It is true that, at the least, the robot must have a way of detecting the goal location. However, explicit localisation with reference to a map is **NOT** the only strategy that qualifies as a goal detector.

An alternative, adopted by the behaviour-based community, suggests that, since sensors and effectors are noisy and information-limited, one should **avoid** creating a geometric map for localisation. Instead, they suggest designing sets of behaviours which together result in the **desired robot motion**.

In its essence, this approach avoids explicit reasoning about localisation and position, and therefore generally avoids explicit path planning as well.

This technique is based on a idea that, there exists a procedural solution to the particular navigation problem at hand. For example, in **Fig.** 2.3, the behavioralist approach to navigating from Room A to Room B might be to design a left-wall-following behavior and a detector for Room B that is triggered by some unique queue in Room B, such as the color of the carpet. Then, the robot can reach Room B by engaging the left wall follower with the Room B detector as the termination condition for the program.

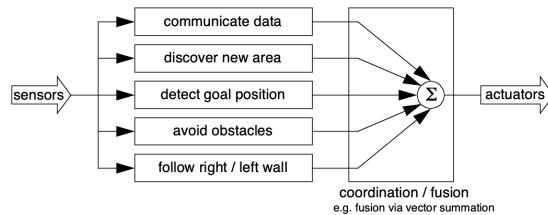


Figure 2.4: An Architecture for Behavior-based Navigation

The architecture of this solution to a specific navigation problem is shown in **Fig. 2.4**. The key advantage of this method is that, when possible, it may be implemented very quickly for a single environment with a small number of goal positions. It suffers from some disadvantages, however.

- The method does not directly scale to other environments or to larger environments. Often, the navigation code is location-specific, and the same degree of coding and debugging is required to move the robot to a new environment.
- The underlying procedures, such as left-wall-follow, must be carefully designed to produce the desired behaviour. This task may be time-consuming and is heavily dependent on the specific robot hardware and environmental characteristics.
- A behaviour-based system may have multiple active behaviors at any one time. Even when individual behaviours are tuned to optimise performance, this fusion and rapid switching between multiple behaviors can negate that fine-tuning. Often, the addition of each new incremental behavior forces the robot designer to re-tune all of the existing behaviors again to ensure that the new interactions with the freshly introduced behavior are all stable

In contrast to the behaviour-based approach, the map-based approach includes both localisation and cognition modules shown in **Fig. 2.5**.

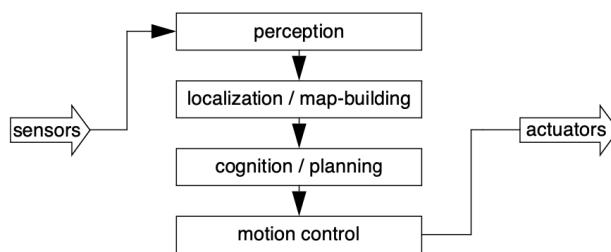


Figure 2.5: An Architecture for Map-based (or model-based) Navigation

In map-based navigation, the robot **explicitly** attempts to localise by collecting sensor data, then updating some belief about its position with respect to a map of the environment. The key advantages of the map-based approach for navigation are as follows:

- The explicit, map-based concept of position makes the system's belief about position transparent.

ently available to the human operators.

- The existence of the map itself represents a medium for communication between human and robot as the human can simply give the robot a new map if the robot goes to a new environment.
- The map, if created by the robot, can be used by humans as well, achieving two uses.

The map-based approach will require more up-front development effort to create a navigating AMR. The hope is that the development effort results in an architecture which can successfully map and navigate a variety of environments, thereby compensating for the up-front design cost over time.

Of course the primary risk of the map-based approach is that an internal representation, rather than the real world itself, is being constructed and trusted by the robot. If that model diverges from reality,¹³ then the robot's behaviour may be undesirable at best or wrong at worst, even if the raw sensor values of the robot are only transiently incorrect.

¹³As in if the robot gets the wrong idea about its environment and draws the wrong map.

In the remainder of this chapter, we focus on a discussion of map-based approaches and, specifically, the localisation component of these techniques. These approaches are particularly appropriate for study given their significant recent successes in enabling AMR to navigate a variety of environments, from academic research buildings to factory floors and museums around the world.

2.4 Representing Belief

The fundamental issue which differentiates different types of map-based localisation systems is the issue of **representation**. There are two (2) specific concepts which the robot must represent, and each has its own unique possible solutions.

- Representation of the environment,
- The map.

What aspects of the environment are contained in this map? At what level of fidelity does the map represent the environment? These are the design questions for map representation.

The robot must also have a representation of its **belief** regarding its position on the map.

Does the robot identify a single unique position as its current position, or does it describe its position in terms of a set of possible positions? If multiple possible positions are expressed in a single belief, how are those multiple positions ranked, if at all?

These are the design questions for belief representation. Decisions along these two (2) design axes can result in varying levels of architectural complexity, computational complexity and overall localisation accuracy.

We will start by discussing belief representation. The first major branch in a taxonomy of belief representation systems differentiates between single hypothesis and multiple hypothesis belief systems.

- The former covers solutions in which the robot postulates its unique position,
- The latter enables a AMR to describe the degree to which it is uncertain about its position.

A sampling of different belief and map representations is shown in figure 5.9.

2.4.1 Single Hypothesis Belief

The single hypothesis belief representation is the most direct possible postulation of an AMR's position [34].

Given some environmental map, the robot's belief about position is expressed as a single unique point on the map.

In **Fig.** 2.6, three (3) examples of a single hypothesis belief are shown using three different map representations of the same actual environment shown in **Fig.** 2.6a. In 5.10b, a single point is geometrically annotated as the robot's position in a continuous two-dimensional geometric map. In 5.10c, the map is a discrete, tessellated map, and the position is noted at the same level of fidelity

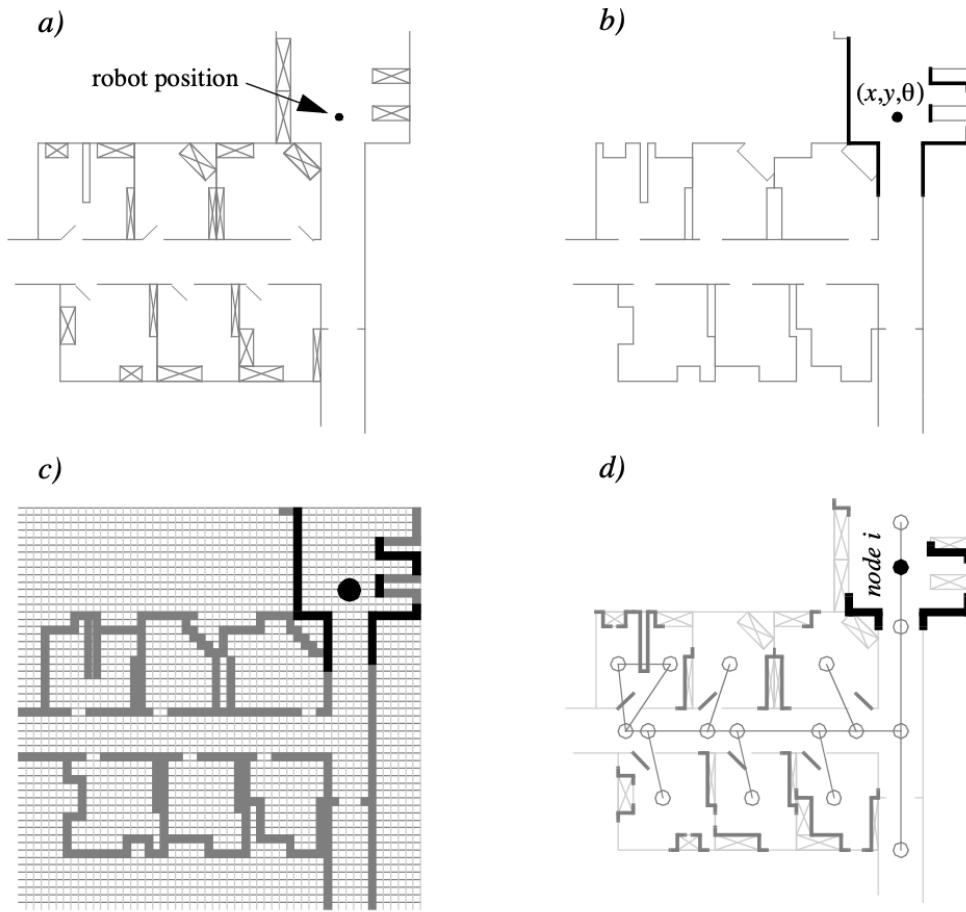


Figure 2.6: The three (3) examples of single hypotheses of position using different map representation. **a)** real map with walls, doors and furniture **b)** line-based map -> around 100 lines with two parameters **c)** occupancy grid based map -> around 3000 grid cells sizing 50x50 cm **d)** topological map using line features (Z/S-lines) and doors -> around 50 features and 18 nodes

as the map cell size. In 5.10d, the map is not geometrical at all but abstract and topological. In this case, the single hypothesis of position involves identifying a single node i in the topological graph as the robot's position.

The principal advantage of the single hypothesis representation of position stems from the fact that, given a unique belief, there is no position ambiguity. The unambiguous nature of this representation facilitates decision-making at the robot's cognitive level (e.g. path planning). The robot can simply assume that its belief is correct, and can then select its future actions based on its unique position.

Just as decision-making is facilitated by a single-position hypothesis, so updating the robot's belief regarding position is also facilitated, since the single position must be updated by definition to a new, single position. The challenge with this position update approach, which ultimately is the principal disadvantage of single-hypothesis representation, is that robot motion often induces uncertainty due to effector and sensory noise.

Forcing the position update process to always generate a single hypothesis of position is challenging and, often, impossible.

2.4.2 Multiple Hypothesis Belief

In the case of multiple hypothesis beliefs regarding position, the robot tracks **NOT** just a single possible position but a possibly **infinite set of positions**. In one simple example originating in the work of Jean-Claude Latombe [5, 89], the robot's position is described in terms of a convex polygon positioned in a two-dimensional map of the environment.

This multiple hypothesis representation communicates the set of possible robot positions geometrically, with no preference ordering over the positions. Each point in the map is simply either contained by the polygon and, therefore, in the robot's belief set, or outside the polygon and thereby excluded. Mathematically, the position polygon serves to partition the space of possible robot positions. Such a polygonal representation of the multiple hypothesis belief can apply to a continuous, geometric map of the environment or, alternatively, to a tessellated, discrete approximation to the continuous environment.

It may be useful, however, to incorporate some ordering on the possible robot positions, capturing the fact that some robot positions are likelier than others. A strategy for representing a continuous multiple hypothesis belief state along with a preference ordering over possible positions is to model the belief as a mathematical distribution. For example, [42,47] note the robot's position belief using an X,Y point in the two-dimensional environment as the mean μ plus a standard deviation parameter σ , thereby defining a Gaussian distribution. The intended interpretation is that the distribution at each position represents the probability assigned to the robot being at that location. This representation is particularly amenable to mathematically defined tracking functions, such as the Kalman Filter, that are designed to operate efficiently on Gaussian distributions.

An alternative is to represent the set of possible robot positions, not using a single Gaussian probability density function, but using discrete markers for each possible position. In this case, each possible robot position is individually noted along with a confidence or probability parameter (See Fig. (5.11)). In the case of a highly tessellated map this can result in thousands or even tens of thousands of possible robot positions in a single belief state.

The key advantage of the multiple hypothesis representation is that the robot can explicitly maintain uncertainty regarding its position. If the robot only acquires partial information regarding position from its sensors and effectors, that information can conceptually be incorporated in an updated belief.

A more subtle advantage of this approach revolves around the robot's ability to explicitly measure its own degree of uncertainty regarding position. This advantage is the key to a class of localisation and navigation solutions in which the robot not only reasons about reaching a particular goal, but reasons about the future trajectory of its own belief state. For instance, a robot may choose paths

that minimise its future position uncertainty. An example of this approach is [90], in which the robot plans a path from point A to B that takes it near a series of landmarks in order to mitigate localisation difficulties. This type of explicit reasoning about the effect that trajectories will have on the quality of localisation requires a multiple hypothesis representation.

One of the fundamental disadvantages of the multiple hypothesis approaches involves decision-making. If the robot represents its position as a region or set of possible positions, then how shall it decide what to do next? Figure 5.11 provides an example. At position 3, the robot's belief state is distributed among 5 hallways separately. If the goal of the robot is to travel down one particular hallway, then given this belief state what action should the robot choose?

The challenge occurs because some of the robot's possible positions imply a motion trajectory that is inconsistent with some of its other possible positions. One approach that we will see in the case studies below is to assume, for decision-making purposes, that the robot is physically at the most probable location in its belief state, then to choose a path based on that current position. But this approach demands that each possible position have an associated probability.

In general, the right approach to such a decision-making problems would be to decide on trajectories that eliminate the ambiguity explicitly. But this leads us to the second major disadvantage of the multiple hypothesis approaches. In the most general case, they can be computationally very expensive. When one reasons in a three dimensional space of discrete possible positions, the number of possible belief states in the single hypothesis case is limited to the number of possible positions in the 3D world. Consider this number to be N . When one moves to an arbitrary multiple hypothesis representation, then the number of possible belief states is the power set of N , which is far larger: 2^N . Thus explicit reasoning about the possible trajectory of the belief state over time quickly becomes computationally untenable as the size of the environment grows. There are, however, specific forms of multiple hypothesis representations that are somewhat more constrained, thereby avoiding the computational explosion while allowing a limited type of multiple hypothesis belief. For example, if one assumes a Gaussian distribution of probability centered at a single position, then the problem of representation and tracking of belief becomes equivalent to Kalman Filtering, a straightforward mathematical process described below. Alternatively, a highly tessellated map representation combined with a limit of 10 possible positions in the belief state, results in a discrete update cycle that is, at worst, only 10x more computationally expensive than single hypothesis belief update.

In conclusion, the most critical benefit of the multiple hypothesis belief state is the ability to maintain a sense of position while explicitly annotating the robot's uncertainty about its own position. This powerful representation has enabled robots with limited sensory information to navigate robustly in an array of environments, as we shall see in the case studies below.

2.5 Representing Maps

The problem of representing the environment in which an AMR moves is a dual of the problem of representing the robot's possible position or positions. Decisions made regarding the environmental representation can have impact on the choices available for robot position representation.

Often the fidelity of the position representation is bounded by the fidelity of the map.

There are three (3) fundamental relationships which must be understood when choosing a particular map representation:

1. The precision of the map must appropriately match the precision with which the robot needs to achieve its goals.
2. The precision of the map and the type of features represented must match the precision and data types returned by the robot's sensors.
3. The complexity of the map representation has direct impact on the computational complexity of reasoning about mapping, localisation and navigation.

Using the aforementioned criteria, we identify and discuss critical design choices in creating a map representation. Each such choice has great impact on the relationships, and on the resulting robot localisation architecture. As we will see, the choice of possible map representations is broad, if not expansive. Selecting an appropriate representation requires understanding all of the trade-offs inherent in that choice as well as understanding the specific context in which a particular AMR implementation must perform localisation.

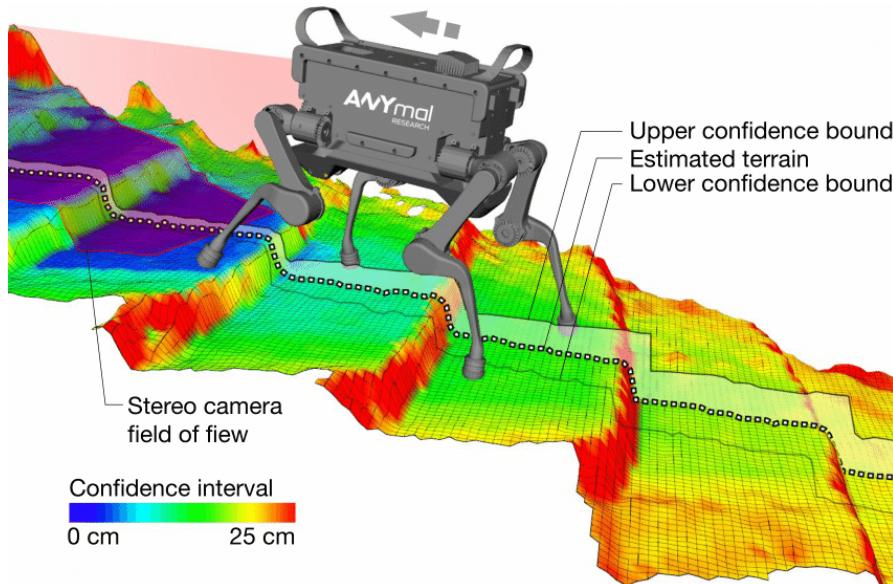


Figure 2.7: The presented robot-centric mapping framework enables mobile robots to create consistent elevation maps of the terrain. Mapping does not necessarily need to be done only in 2D as robots which will be used in outdoor environment would need the height of the map as well [35].

2.5.1 Continuous Representation

A continuous-valued map is one method for **exact** decomposition of the environment. The position of environmental features can be mapped precisely in continuous space.

AMR implementations to date use continuous maps only in two (2) dimensional representations, as increasing the number of dimensions can result in high computational load on the AMR navigation computer.

A common approach is to combine the exactness of a continuous representation with the compactness of the closed world assumption. This means that one assumes the representation will specify all environmental objects in the map, and that any area in the map which is devoid of objects has no objects in the corresponding portion of the environment. Therefore, the total storage needed in the map is proportional to the density of objects in the environment, and a sparse environment can be represented by a low-memory map.

One example of such a representation, shown in **Fig. 2.8**, is a 2D representation in which polygons represent all obstacles in a continuous-valued coordinate space. This is similar to the method used by Latombe [5, 113] and others to represent environments for AMR path planning techniques. In the case of [5, 113], most of the experiments are in fact simulations run exclusively within the computer's memory. Therefore, no real effort would have been expended to attempt to use sets of polygons to describe a real-world environment, such as a park or office building.

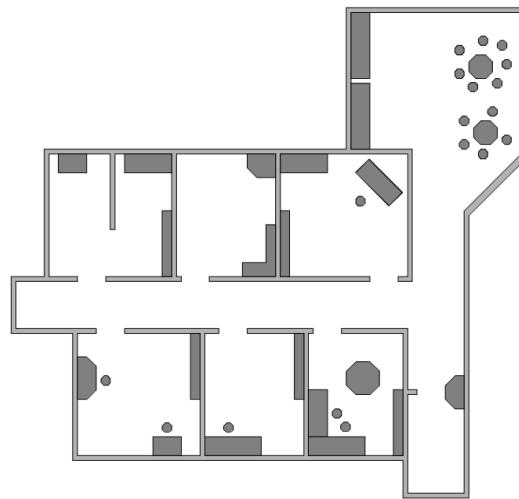


Figure 2.8: A continuous representation using polygons as environmental obstacles.

In other work in which real environments must be captured by the maps, there seems to be a trend towards **selectivity** and **abstraction**. The human map-maker tends to capture on the map, for localisation purposes, only objects that can be detected by the robot's sensors and, furthermore, only a subset of the features of real-world objects.

It should be immediately apparent that geometric maps can capably represent the physical locations of objects without referring to their texture, colour, elasticity, or any other such secondary features that do not relate directly to position and space.

In addition to this level of abstraction, an AMR map can further reduce memory usage by capturing only **aspects of object geometry** which are **immediately relevant** to localisation. For example all objects may be approximated using very simple convex polygons,¹⁴ sacrificing map felicity for the sake of computational speed.

One excellent example involves **line extraction**. Many indoor AMR rely upon laser range-finding devices to recover distance readings to nearby objects. Such robots can automatically extract best-fit lines from the dense range data provided by thousands of points of laser strikes. Given such a line extraction sensor, an appropriate continuous mapping approach is to populate the map with a set of infinite lines. The continuous nature of the map guarantees that lines can be positioned at arbitrary positions in the plane and at arbitrary angles. The abstraction of real environmental objects such as walls and intersections captures only the information in the map representation that matches the type of information recovered by the AMR's rangefinding sensor.

¹⁴A convex polygon is any shape that has all interior angles that measure less than 180 degrees

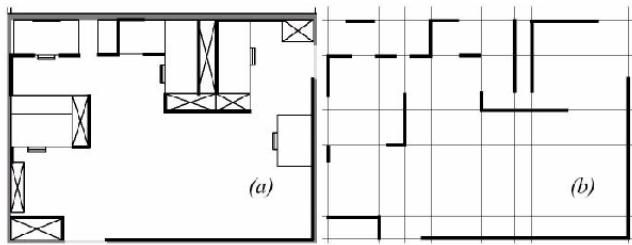


Figure 2.9: Example of a continuous-valued line representation of EPFL. left: real map right: representation with a set of infinite lines.

Fig. 2.9 shows a map of an indoor environment at EPFL using a continuous line representation. Note that the only environmental features captured by the map are straight lines, such as those found at corners and along walls. This represents not only a sampling of the real world of richer features, but also a simplification, for an actual wall may have texture and relief that is not captured by the mapped line. The impact of continuous map representations on position representation is primarily positive. In the case of single hypothesis position representation, that position may be specified as any continuous-valued point in the coordinate space, and therefore extremely high accuracy is possible. In the case of multiple hypothesis position representation, the continuous map enables two types of multiple position representation. In one case, the possible robot position may be depicted as a geometric shape in the hyperplane, such that the robot is known to be within the bounds of that shape. This is shown in Figure 5.30, in which the position of the robot is depicted by an oval bounding area. Yet, the continuous representation does not disallow representation of position in the form of a discrete set of possible positions. For instance, in [111] the robot position belief state is captured by sampling nine continuous-valued positions from within a region near the robot's best known position. This algorithm captures, within a continuous space, a discrete sampling of possible robot positions. In summary, the key advantage of a continuous map representation is the potential for high accuracy and expressiveness with respect to the environmental configuration as well as

the robot position within that environment. The danger of a continuous representation is that the map may be computationally costly. But this danger can be tempered by employing abstraction and capturing only the most relevant environmental features. Together with the use of the closed world assumption, these techniques can enable a continuous-valued map to be no more costly, and sometimes even less costly, than a standard discrete representation.

2.5.2 Decomposition Methods

In previous section, we discussed one method of simplification, in which the continuous map representation contains a set of infinite lines which approximate real-world environmental lines based on a two-dimensional slice of the world.

Basically this transformation from the real world to the map representation is a filter that removes all non-straight data and furthermore extends line segment data into infinite lines that require fewer parameters.

A more dramatic form of simplification is abstraction:

a general decomposition and selection of environmental features.

In this section, we explore decomposition as applied in its more extreme forms to the question of map representation. Why would one radically decompose the real environment during the design of a map representation? The immediate disadvantage of decomposition and abstraction is the loss of fidelity between the map and the real world. Both qualitatively, in terms of overall structure, and quantitatively, in terms of geometric precision, a highly abstract map does not compare favourably to a high-fidelity map.

Despite this disadvantage, decomposition and abstraction may be useful if the abstraction can be planned carefully so as to capture the relevant, useful features of the world while discarding all other features. The advantage of this approach is that the map representation can potentially be minimised. Furthermore, if the decomposition is hierarchical, such as in a pyramid of recursive abstraction, then reasoning and planning with respect to the map representation may be computationally far superior to planning in a fully detailed world model.

A standard, lossless form of opportunistic decomposition is termed exact cell decomposition. This method, introduced by [5], achieves decomposition by selecting boundaries between discrete cells based on geometric criticality.

Figure 5.14 depicts an exact decomposition of a planar workspace populated by polygonal obstacles. The map representation tessellates the space into areas of free space. The representation can be extremely compact because each such area is actually stored as a single node, shown in the graph at the bottom of Figure 5.14.

The underlying assumption behind this decomposition is that the particular position of a robot within

each area of free space does not matter. What matters is the robot's ability to traverse from each area of free space to the adjacent areas. Therefore, as with other representations we will see, the resulting graph captures the adjacency of map locales. If indeed the assumptions are valid and the robot does not care about its precise position within a single area, then this can be an effective representation that nonetheless captures the connectivity of the environment.

Such an exact decomposition is not always appropriate. Exact decomposition is a function of the particular environment obstacles and free space. If this information is expensive to collect or even unknown, then such an approach is not feasible.

An alternative is fixed decomposition, in which the world is tessellated, transforming the continuous real environment into a discrete approximation for the map. Such a transformation is demonstrated in Figure 5.15, which depicts what happens to obstacle-filled and free areas during this transformation. The key disadvantage of this approach stems from its inexact nature. It is possible for narrow passageways to be lost during such a transformation, as shown in Figure 5.15. Formally this means that fixed decomposition is sound but not complete. Yet another approach is adaptive cell decomposition as presented in Figure 5.16.

The concept of fixed decomposition is extremely popular in AMRics; it is perhaps the single most common map representation technique currently utilised. One very popular

version of fixed decomposition is known as the occupancy grid representation [91]. In an occupancy grid, the environment is represented by a discrete grid, where each cell is either filled (part of an obstacle) or empty (part of free space). This method is of particular value when a robot is equipped with range-based sensors because the range values of each sensor, combined with the absolute position of the robot, can be used directly to update the filled/empty value of each cell. In the occupancy grid, each cell may have a counter, whereby the value 0 indicates that the cell has not been "hit" by any ranging measurements and, therefore, it is likely free space. As the number of ranging strikes increases, the cell's value is incremented and, above a certain threshold, the cell is deemed to be an obstacle. By discounting the values of cells over time, both hysteresis and the possibility of transient obstacles can be represented using this occupancy grid approach. Figure 5.17 depicts an occupancy grid representation in which the darkness of each cell is proportional to the value of its counter. One commercial robot that uses a standard occupancy grid for mapping and navigation is the Cye robot [112].

There remain two main disadvantages of the occupancy grid approach. First, the size of the map in robot memory grows with the size of the environment and, if a small cell size is used, this size can quickly become untenable. This occupancy grid approach is not compatible with the closed world assumption, which enabled continuous representations to have potentially very small memory requirements in large, sparse environments. In contrast, the occupancy grid must have memory set aside for every cell in the matrix. Furthermore, any fixed decomposition method such as this imposes a geometric grid on the world *a priori*, regardless of the environmental details. This can be inappropriate in cases where geometry is not the most salient feature of the environment. For these reasons, an alternative, called topological decomposition, has been the subject of some exploration

in AMRics. Topological approaches avoid direct measurement of geometric environmental qualities, instead concentrating on characteristics of the environment that are most relevant to the robot for localisation. Formally, a topological representation is a graph that specifies two things: nodes and the connectivity between those nodes. Insofar as a topological representation is intended for the use of a AMR, nodes are used to denote areas in the world and arcs are used to denote adjacency of pairs of nodes. When an arc connects two nodes, then the robot can traverse from one node to the other without requiring traversal of any other intermediary node. Adjacency is clearly at the heart of the topological approach, just as adjacency in a cell decomposition representation maps to geometric adjacency in the real world. However, the topological approach diverges in that the nodes are not of fixed size nor even specifications of free space. Instead, nodes document an area based on any sensor discriminant such that the robot can recognise entry and exit of the node. Figure 5.18 depicts a topological representation of a set of hallways and offices in an indoor

environment. In this case, the robot is assumed to have an intersection detector, perhaps using sonar and vision to find intersections between halls and between halls and rooms. Note that nodes capture geometric space and arcs in this representation simply represent connectivity. Another example of topological representation is the work of Dudek [49], in which the goal is to create a AMR that can capture the most interesting aspects of an area for human consumption. The nodes in Dudek's representation are visually striking locales rather than route intersections. In order to navigate using a topological map robustly, a robot must satisfy two constraints. First, it must have a means for detecting its current position in terms of the nodes of the topological graph. Second, it must have a means for traveling between nodes using robot motion. The node sizes and particular dimensions must be optimised to match the sensory discrimination of the AMR hardware. This ability to "tune" the representation to the robot's particular sensors can be an important advantage of the topological approach. However, as the map representation drifts further away from true geometry, the expressiveness of the representation for accurately and precisely describing a robot position is lost. Therein lies the compromise between the discrete cell-based map representations and the topological representations. Interestingly, the continuous map representation has the potential to be both compact like a topological representation and precise as with all direct geometric representations. Yet, a chief motivation of the topological approach is that the environment may contain important non-geometric features - features that have no ranging relevance but are useful for localisation. In Chapter 4 we described such whole-image vision-based features. In contrast to these whole-image feature extractors, often spatially localised landmarks are artificially placed in an environment to impose a particular visual-topological connectivity upon the environment. In effect, the artificial landmark can impose artificial structure. Examples of working systems operating with this landmark-based strategy have also demonstrated success. Latombe's landmark-based navigation research [89] has been implemented on real-world indoor AMRs that employ paper landmarks attached to the ceiling as the locally observable features. Chips the museum robot is another robot that uses man-made landmarks to obviate the localisation problem. In this case, a bright pink square serves as a landmark with dimensions and color signature that would be hard to accidentally reproduce in a museum environment [88]. One such museum landmark is shown in Figure (5.19). In summary, range is clearly not the only measurable and useful environmental value for a AMR. This is particularly true due to the advent of color vision as well as

laser rangefinding, which provides reflectance information in addition to range information. Choosing a map representation for a particular AMR requires first understanding the sensors available on the AMR and second understanding the AMR's functional requirements (e.g. required goal precision and accuracy).

2.5.3 Current Challenges

Previous section describe major design decisions with regards to map representation choices. There are, however, fundamental real-world features which AMR map representations do not work as well. These continue to be the subject of open research, and several such challenges are described below.

The real world is **dynamic**. As AMRs come to work and move in the same spaces as humans, they will encounter:

- moving people,
- cars,
- strollers, and
- transient obstacles.

This is particularly true when one considers a home setting with which domestic robots will someday need to contend.

The map representations described previously do not, in general, have **explicit methods** for identifying and distinguishing between permanent obstacles (e.g. walls, doorways, etc.) and transient obstacles (e.g., humans, shipping packages, etc.). The current state of the art in terms of AMR sensors is partly to blame for this shortcoming. Although vision research is rapidly advancing, robust sensors that discriminate between moving animals and static structures from a moving reference frame are not yet available. Furthermore, estimating the motion vector of transient objects remains a research problem.

Usually, the assumption behind the above map representations is that all objects on the map are effectively **static**. Partial success can be achieved by discounting mapped objects over time. For example, occupancy grid techniques can be more robust to dynamic settings by introducing temporal discounting, effectively treating transient obstacles as noise. The more challenging process of map creation is particularly fragile to environment dynamics; most mapping techniques generally require that the environment be free of moving objects during the mapping process. One exception to this limitation involves topological representations. Because precise geometry is not important, transient objects have little effect on the mapping or localisation process, subject to the critical constraint that the transient objects must not change the topological connectivity of the environment. Still, neither the occupancy grid representation nor a topological approach is actively recognizing and representing transient objects as distinct from both sensor error and permanent map features.

As vision sensing provides more robust and more informative content regarding the transience and motion details of objects in the world, researchers will in time propose representations that make use of that information. A classic example involves occlusion by human crowds. Museum tour guide robots¹⁵ generally suffer from an extreme amount of occlusion. If the robot's sensing suite is located along the robot's body, then the robot is effectively blind when a group of human visitors completely surrounds the robot. This is because its map contains only environment features that are, at that point, fully hidden from the robot's sensors by the wall of people. In the best case, the robot should recognise its occlusion and make no effort to localise using these invalid sensor readings. In the worst case, the robot will localise with the fully occluded data, and will update its location incorrectly. A vision sensor that can discriminate the local conditions of the robot (e.g. we are surrounded by people) can help eliminate this error mode.



¹⁵An Example of a museum tour guide robot used in the National Museum of Korea [36].

A second open challenge in AMR localisation involves the traversal of open spaces. Existing localisation techniques generally depend on local measures such as range, thereby demanding environments that are somewhat densely filled with objects that the sensors can detect and measure. Wide open spaces such as parking lots, fields of grass and indoor open-spaces such as those found in convention centres or expos pose a difficulty for such systems due to their relative sparseness. Indeed, when populated with humans, the challenge is exacerbated because any mapped objects are almost certain to be occluded from view by the people.

Once again, more recent technologies provide some hope for overcoming these limitations. Both vision and state-of-the-art laser range-finding devices offer outdoor performance with ranges of up to a hundred meters and more. Of course, GPS performs even better. Such long-range sensing may be required for robots to localise using distant features.

This trend teases out a hidden assumption underlying most topological map representations. Usually, topological representations make assumptions regarding spatial locality:

a node contains objects and features that are themselves within that node.

The process of map creation therefore involves making nodes which are, in their own self-contained way, recognizable by virtue of the objects contained within the node. Therefore, in an indoor environment, each room can be a separate node. This is a reasonable assumption as each room will have a layout and a set of belongings that are **unique** to that room.

However, consider the outdoor world of a wide-open park.

Where should a single node end and the next node begin?

The answer is unclear as objects which are far away from the current node, or position, can give information for the localisation process. For example, the hump of a hill at the horizon, the position of a river in the valley and the trajectory of the Sun all are non-local features that have great bearing on one's ability to infer current position.

The spatial locality assumption is violated and, instead, replaced by a visibility criterion:

the node or cell may need a mechanism for representing objects that are measurable and visible from that cell.

Once again, as sensors and outdoor locomotion mechanisms improve, there will be greater urgency to solve problems associated with localisation in wide-open settings, with and without GPS-type global localisation sensors.¹⁶

We end this section with one final open challenge that represents one of the fundamental academic research questions of robotics: **sensor fusion**.

Information: Sensor Fusion

A variety of measurement types are possible using off-the-shelf robot sensors, including heat, range, acoustic and light-based reflectivity, color, texture, friction, etc. Sensor fusion is a research topic closely related to map representation. Just as a map must embody an environment in sufficient detail for a robot to perform localisation and reasoning, sensor fusion demands a representation of the world that is sufficiently general and expressive that a variety of sensor types can have their data correlated appropriately, strengthening the resulting percepts well beyond that of any individual sensor's readings.

An implementation example implementation of sensor fusion to date is that of neural network classifier. Using this technique, any number and any type of sensor values may be jointly combined in a network that will use whatever means necessary to optimise its classification accuracy. For the AMR that must use a human-readable internal map representation, no equally general sensor fusion scheme has yet been born. It is reasonable to expect that, when the sensor fusion problem is solved, integration of a large number of disparate sensor types may easily result in sufficient discriminatory power for robots to achieve real-world navigation, even in wide-open and dynamic circumstances such as a public square filled with people.

¹⁶Of course with the use of a GNSS, the localisation problem may completely be solved, however in cost saving measures one would wish to avoid the use of them as they can be expensive.

2.6 Probabilistic Map-Based Localisation

2.6.1 Introduction

As stated previously, multiple hypothesis position representation is advantageous because the robot can explicitly track its own beliefs regarding its possible positions in the environment. Ideally, the robot's belief state will change, over time, as is consistent with its motor outputs and perceptual inputs. One geometric approach to multiple hypothesis representation, mentioned earlier, involves identifying the possible positions of the robot by specifying a polygon in the environmental representation [113]. This method does not provide any indication of the relative chances between various possible robot positions. Probabilistic techniques differ from this because they explicitly identify probabilities with the possible robot positions, and for this reason these methods have been the focus of recent research. In the following sections we present two classes of probabilistic localisation. The first class, Markov localisation, uses an explicitly specified probability distribution across all possible robots positions. The second method, Kalman filter localisation, uses a Gaussian probability density representation of robot position and scan matching for localisation. Unlike Markov localisation, Kalman filter localisation does not independently consider each possible pose in the robot's configuration space. Interestingly, the Kalman filter localization process results from the Markov localisation axioms if the robot's position uncertainty is assumed to have a Gaussian form [28 page 43-44]. Before discussing each method in detail, we present the general robot localisation problem and solution strategy. Consider a AMR moving in a known environment. As it starts to move, say from a precisely known location, it can keep track of its motion using odometry. Due to odometry uncertainty, after some movement the robot will become very uncertain about its position (see section 5.2.4). To keep position uncertainty from growing unbounded, the robot must localise itself in relation to its environment map. To localise, the robot might use its on-board sensors (ultrasonic, range sensor, vision) to make observations of its environment. The information provided by the robot's odometry, plus the information provided by such exteroceptive observations can be combined to enable the robot to localise as well as possible with respect to its map. The processes of updating based on proprioceptive sensor values and exteroceptive sensor values are often separated logically, leading to a general two-step process for robot position update. Action update represents the application of some action model Act to the AMR's proprioceptive encoder measurements o and prior belief state s to yield a new belief s' representing the robot's belief about its current position. Note that throughout this chapter we will assume that the robot's proprioceptive encoder measurements are used as the best possible measure of its actions over time. If, for instance, a differential drive robot had motors without encoders connected to its wheels and employed open-loop control, then instead of encoder measurements the robot's highly uncertain estimates of wheel spin would need to be incorporated. We ignore such cases and therefore have a simple formula:

$$s'_t = \text{Act}(o_t, s_{t-1}) \quad (2.1)$$

Perception update represents the application of some perception model See to the AMR's exteroceptive sensor inputs i and updated belief state s' to yield a refined belief s'' representing the

robot's current position:

$$s_t = \text{See} \left(i_t, s'_{t-1} \right) \quad (2.2)$$

The perception model See and sometimes the action model Act are abstract functions of both the map and the robot's physical configuration.¹⁷

¹⁷such as sensors and their positions, kinematics, etc.

In general, the action update process **contributes uncertainty** to the robot's belief about position:

encoders have error and therefore motion is somewhat nondeterministic.

In contrast, perception update generally **refines** the belief state. Sensor measurements, when compared to the robot's environmental model, tend to provide clues regarding the robot's possible position.

In the case of Markov localisation, the robot's belief state is usually represented as separate probability assignments for every possible robot pose in its map. The action update and perception update processes must update the probability of every cell in this case. Kalman filter localisation represents the robot's belief state using a single, well-defined Gaussian probability density function, and therefore retains just a μ and σ parameterisation of the robot's belief about position with respect to the map. Updating the parameters of the Gaussian distribution is all that is required. This fundamental difference in the representation of belief state leads to the following advantages and disadvantages of the two (2) methods, as presented in [37]:

- Markov localization allows for localization starting from any unknown position and can thus recover from ambiguous situations because the robot can track multiple, completely disparate possible positions. However, to update the probability of all positions within the whole state space at any time requires a discrete representation of the space (grid). The required memory and computational power can thus limit precision and map size.
- Kalman filter localization tracks the robot from an initially known position and is inherently both precise and efficient. In particular, Kalman filter localization can be used in continuous world representations. However, if the uncertainty of the robot becomes too large (e.g. due to a robot collision with an object) and thus not truly unimodal, the Kalman filter can fail to capture the multitude of possible robot positions and can become irrevocably lost.

Improvements are achieved or proposed by either only updating the state space of interest within the Markov approach [38] or by combining both methods to create a hybrid localization system [37].

We will now look at them in great detail.

2.6.2 Markov Localisation

Markov localization tracks the robot's belief state using an arbitrary probability density function to represent the robot's position. In practice, all known Markov localization systems implement this generic belief representation by first tessellating the robot configuration space into a finite, discrete number of possible robot poses in the map. In actual applications, the number of possible poses can range from several hundred positions to millions of positions.

Given such a generic conception of robot position, a powerful update mechanism is required that can compute the belief state that results when new information (e.g. encoder values and sensor values) is incorporated into a prior belief state with arbitrary probability density. The solution is born out of probability theory, and so the next section describes the foundations of probability theory that apply to this problem, notably Bayes formula. Then, two subsequent subsections provide case studies, one robot implementing a simple feature-driven topological representation of the environment [39], and the other using a geometric grid-based map [38].

Application of Probability for Localisation

Given a discrete representation of robot positions, to express a belief state we wish to assign to each possible robot position a probability that the robot is indeed at that position.

From probability theory we use the term $P(A)$ to denote the probability that A is true. This is also called the prior probability of A because it measures the probability that A is true independent of any additional knowledge we may have.

For example we can use $P(r_t = l)$ to denote the prior probability that the robot r is at position l at time t .

In practice, we wish to compute the probability of each individual robot position given the encoder and sensor evidence the robot has collected. For this, we use the term $P(A|B)$ to denote the **conditional probability** of A given that we know B .

For example, we use $P(r_t = l|i_t)$ to denote the probability that the robot is at position l given that the robot's sensor inputs i .

The question is,

how can a term such as $P(r_t = l|i_t)$ be simplified to its constituent parts so that it can be computed?

The answer lies in the product rule, which states:

$$P(A \wedge B) = P(A|B) P(B) \quad (2.3)$$

The equation given in Eq. (2.3) is relatively straightforward, as the probability of both A and¹⁸ B being true is being related to B being true and the other being conditionally true. But you should be able to convince yourself that the alternate equation is equally correct:

$$P(A \wedge B) = P(B|A) P(A) \quad (2.4)$$

¹⁸To simplify notation we will be using the wedge (\wedge) symbol to denote AND, and the vee (\vee) symbol to denote OR.

Using both Eq. (2.3) and Eq. (2.4) together, we can derive Bayes formula for computing $P(A|B)$:

$$P(A|B) = \frac{P(B|A) P(A)}{P(B)} \quad (2.5)$$

We use Bayes rule to compute the robot's new belief state as a function of its sensory inputs and its former belief state. But to do this properly, we must recall the basic goal of the Markov localisation approach:

a discrete set of possible robot positions L are represented.

The belief state of the robot must assign a probability $P(r_t = l)$ for each location l in L .

The See function described in Eq. (2.2) expresses a mapping from a belief state and sensor input to a refined belief state. To do this, we must update the probability associated with each position l in L , and we can do this by directly applying Bayes formula to every such l .

In denoting this, we will stop representing the temporal index t for simplicity and will further use $P(l)$ to mean $P(r = l)$:

$$P(l|i) = \frac{P(i|l) P(l)}{P(i)} \quad (2.6)$$

The value of $P(l|i)$ is key to Eq. (2.6), and this probability of a sensor input at each robot position must be computed using some model. An obvious strategy would be to consult the robot's map, identifying the probability of particular sensor readings with each possible map position, given knowledge about the robot's sensor geometry and the mapped environment. The value of $P(l)$ is easy to recover in this case. It is simply the probability $P(r = l)$ associated with the belief state before the perceptual update process.

Finally, note that the denominator $P(i)$ does **NOT** depend upon l ; that is, as we apply Eq. (2.6) to all positions l in L , the denominator never varies.

Because it is effectively constant, in practice this denominator is usually dropped and, at the end of the perception update step, all probabilities in the belief state are re-normalized to sum at 1.0.

Now consider the Act function of Eq. (2.1). Act maps a former belief state and encoder measurement (i.e. robot action) to a new belief state. To compute the probability of position l in the new belief state, one must integrate over all the possible ways in which the robot may have reached l according

to the potential positions expressed in the former belief state. This is subtle but fundamentally important. The same location l can be reached from multiple source locations with the same encoder measurement o because the encoder measurement is uncertain. Temporal indices are required in this update equation:

$$P(l_t|o_t) = \int P(l_t|l'_{t-1}, o_t) P(l'_{t-1}) dl'_{t-1} \quad (2.7)$$

Thus, the total probability for a specific position l is built up from the individual contributions from every location l' in the former belief state given encoder measurement o . Equations 5.21 and 5.22 form the basis of Markov localization, and they incorporate the Markov assumption. Formally, this means that their output is a function only of the robot's previous state and its most recent actions (odometry) and perception. In a general, non-Markovian situation, the state of a system depends upon all of its history. After all, the value of a robot's sensors at time t do not really depend only on its position at time t . They depend to some degree on the trajectory of the robot over time; indeed on the entire history of the robot. For example, the robot could have experienced a serious collision recently that has biased the sensor's behavior. By the same token, the position of the robot at time t does not really depend only on its position at time $t-1$ and its odometric measurements. Due to its history of motion, one wheel may have worn more than the other, causing a left-turning bias over time that affects its current position. So the Markov assumption is, of course, not a valid assumption. However the Markov assumption greatly simplifies tracking, reasoning and planning and so it is an approximation that continues to be extremely popular in mobile robotics.

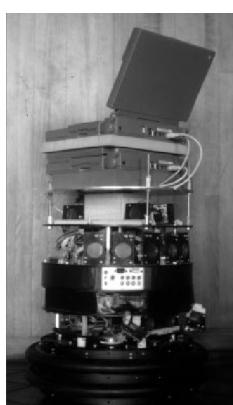
Application: Markov Localisation using a Topological Map

A straightforward application of Markov localization is possible when the robot's environment representation already provides an appropriate decomposition. This is the case when the environment representation is purely topological.

Consider a contest in which each robot is to receive a topological description of the environment. The description would describe only the connectivity of hallways and rooms, with no mention of geometric distance. In addition, this supplied map would be imperfect, containing several false arcs (e.g. a closed door). Such was the case for the 1994 AAAI National Robot Contest, at which each robot's mission was to use the supplied map and its own sensors to navigate from a chosen starting position to a target room.

Dervish¹⁹, the winner of this contest, employed probabilistic Markov localization and used just this multiple hypothesis belief state over a topological environmental representation. We now describe Dervish as an example of a robot with a topological representation and a probabilistic localization algorithm.

Dervish, shown in Figure 5.20, includes a sonar arrangement custom-designed for the 1994 AAAI National Robot Contest. The environment in this contest consisted of a rectilinear indoor office space filled with real office furniture as obstacles. Traditional sonars are arranged radially around the robot in a ring. Robots with such sensor configurations are subject to both tripping over short objects below the ring and to decapitation by tall objects (such as ledges, shelves and tables) that



¹⁹

are above the ring. Dervish's answer to this challenge was to arrange one pair of sonars diagonally upward to detect ledges and other overhangs. In addition, the diagonal sonar pair also proved to ably detect tables, enabling the robot to avoid wandering underneath tall tables. The remaining sonars were clustered in sets of sonars, such that each individual transducer in the set would be at a slightly varied angle to minimize specularity. Finally, two sonars near the robot's base were able to detect low obstacles such as paper cups on the floor.

We have already noted that the representation provided by the contest organizers was purely topological, noting the connectivity of hallways and rooms in the office environment. Thus, it would be appropriate to design Dervish's perceptual system to detect matching perceptual events: the detection and passage of connections between hallways and offices.

This abstract perceptual system was implemented by viewing the trajectory of sonar strikes to the left and right sides of Dervish over time. Interestingly, this perceptual system would use time alone and no concept of encoder value in order to trigger perceptual events. Thus, for instance, when the robot detects a 7 to 17 cm indentation in the width of the hallway for more than one second continuously, a closed door sensory event is triggered. If the sonar strikes jump well beyond 17 cm for more than one second, an open door sensory event triggers.

Sonars have a notoriously problematic error mode known as specular reflection: when the sonar unit strikes a flat surface at a shallow angle, the sound may reflect coherently away from the transducer, resulting in a large overestimate of range. Dervish was able to filter such potential noise by tracking its approximate angle in the hallway and completely suppressing sensor events when its angle to the hallway parallel exceeded 9 degrees. Interestingly, this would result in a conservative perceptual system that would easily miss features because of this suppression mechanism, particularly when the hallway is crowded with obstacles that Dervish must negotiate. Once again, the conservative nature of the perceptual system, and in particular its tendency to issue false negatives, would point to a probabilistic solution to the localization problem so that a complete trajectory of perceptual inputs could be considered.

Dervish's environment representation was a classical topological map, identical in abstraction and information to the map provided by the contest organizers. Figure 5.21 depicts a geometric representation of a typical office environment and the topological map for the same office environment. One can place nodes at each intersection and in each room, resulting in the case of figure 5.21 with four nodes total.

Once again, though, it is crucial that one maximize the information content of the representation based on the available percepts. This means reformulating the standard topological graph shown in Figure 5.21 so that transitions into and out of intersections may both be used for position updates. Figure 5.22 shows a modification of the topological map in which just this step has been taken. In this case, note that there are 7 nodes in contrast to 4. In order to represent a specific belief state, Dervish associated with each topological node n a probability that the robot is at a physical position within the boundaries of n : $p(r = n) \cdot t$. As will become clear below, the probabilistic update used by Dervish was approximate, therefore technically one should refer to the resulting values as likelihoods

rather than prob- abilities.

The perception update process for Dervish functions precisely as in Equation (5.21). Per- ceptual events are generated asynchronously, each time the feature extractor is able to recognize a large-scale feature (e.g. doorway, intersection) based on recent ultrasonic values. Each perceptual event consists of a percept-pair (a feature on one side of the robot or two features on both sides).

	Wall	Closed Door	Open Door	Open Hallway	Foyer
Nothing Detected	0.70	0.40	0.05	0.001	0.30
Closed Door Detected	0.30	0.60	0	0	0.05
Open Door Detected	0	0	0.90	0.10	0.15
Closed Hallway Detected	0	0	0.001	0.90	0.5

Table 2.1: The certainty matrix for the robot [40].

Given a specific percept pair i , Equation (5.21) enables the likelihood of each possible position n to be updated using the formula:

$$P(n|i) = P(i|n) \quad (2.8)$$

The value of $p(n)$ is already available from the current belief state of Dervish, and so the challenge lies in computing $p(i|n)$. The key simplification for Dervish is based upon the realization that, because the feature extraction system only extracts 4 total features and because a node contains (on a single side) one of 5 total features, every possible combination of node type and extracted feature can be represented in a 4×5 table. Dervish's certainty matrix (show in Table 5.1) is just this lookup table. Dervish makes the simplifying assumption that the performance of the feature detector (i.e. the probability that it is correct) is only a function of the feature extracted and the actual feature in the node. With this assumption in hand, we can populate the certainty matrix with confidence estimates for each possible pairing of perception and node type. For each of the five world features that the robot can encounter (wall, closed door, open door, open hallway and foyer) this matrix assigns a likelihood for each of the three one-sided percepts that the sensory system can issue. In addition, this matrix assigns a likelihood that the sensory system will fail to issue a perceptual event altogether (nothing detected).

For example, using the specific values in Table 5.1, if Dervish is next to an open hallway, the likelihood of mistakenly recognizing it as an open door is 0.10. This means that for any node n that is of type Open Hallway and for the sensor value $i=\text{Open door}$, $p(i|n) = 0.10$. Together with a specific topological map, the certainty matrix enables straightforward computation of $p(i|n)$ during the perception update process.

For Dervish's particular sensory suite and for any specific environment it intends to navigate, humans generate a specific certainty matrix that loosely represents its perceptual confidence, along

with a global measure for the probability that any given door will be closed versus opened in the real world.

Recall that Dervish has no encoders and that perceptual events are triggered asynchronously by the feature extraction processes. Therefore, Dervish has no action update step as depicted by Equation (5.22). When the robot does detect a perceptual event, multiple perception up- date steps will need to be performed in order to update the likelihood of every possible robot position given Dervish's former belief state. This is because there is often a chance that the robot has traveled multiple topological nodes since its previous perceptual event (i.e. false negative errors). Formally, the perception update formula for Dervish is in reality a combination of the general form of action update and perception update. The likelihood of position n given perceptual event i is calculated as in Equation (5.22):

$$P(I_t|o_t) = \int P(I_t|I'_{t-1}, o_t) P(I'_{t-1}) dI'_{t-1} \quad (2.9)$$

The value of $p(n')$ denotes the likelihood of Dervish being at position n' as represented by Dervish's former belief state. The temporal subscript $t-i$ is used in lieu of $t-1$ because for each possible position n' the discrete topological distance from n' to n can vary depending on the specific topological map. The calculation of $p(n'|n, i)$ is performed by multiplying the probability of generating perceptual event i at position n by the probability of having failed to generate perceptual events at all nodes between n' and n :

For example (figure 5.23), suppose that the robot has only two nonzero nodes in its belief state, 1-2, 2-3, with likelihoods associated with each possible position: $p(1-2) = 1.0$ and $p(2-3) = 0.2$. For simplicity assume the robot is facing East with certainty. Note that the likelihoods for nodes 1-2 and 2-3 do not sum to 1.0. These values are not formal probabilities, and so computational effort is minimized in Dervish by avoiding normalization altogether. Now suppose that a perceptual event is generated: the robot detects an open hallway on its left and an open door on its right simultaneously. State 2-3 will progress potentially to states 3, 3-4 and 4. But states 3 and 3-4 can be eliminated because the likelihood of detecting an open door when there is only wall is zero. The likelihood of reaching state 4 is the product of the initial likelihood for state 2-3, 0.2, the likelihood of not detecting anything at node 3, (a), and the likelihood of detecting a hallway on the left and a door on the right at node 4, (b). Note that we assume the likelihood of detecting nothing at node 3-4 is 1.0 (a simplifying approximation). (a) occurs only if Dervish fails to detect the door on its left at node 3 (either closed or open), $[(0.6)(0.4) + (1-0.6)(0.05)]$, and correctly detects nothing on its right, 0.7. (b) occurs if Dervish correctly identifies the open hallway on its left at node 4, 0.90, and mis- takes the right hallway for an open door, 0.10. The final formula, $(0.2)[(0.6)(0.4)+(0.4)(0.05)](0.7)[(0.9)(0.1)]$, yields a likelihood of 0.003 for state 4. This is a partial result for $p(4)$ following from the prior belief state node 2-3. Turning to the other node in Dervish's prior belief state, 1-2 will potentially progress to states 2, 2-3, 3, 3-4 and 4. Again, states 2-3, 3 and 3-4 can all be eliminated since the likelihood of detecting an open door when a wall is present is zero. The likelihood of state 2 is the product of the prior likelihood for state 1-2, (1.0), the likelihood of detecting the door on the right as an open door, $[(0.6)(0) + (0.4)(0.9)]$, and the

likelihood of correctly detecting an open hallway to the left, 0.9. The likelihood for being at state 2 is then $(1.0)(0.4)(0.9)(0.9) = 0.3$. In addition, 1-2 progresses to state 4 with a certainty factor of -6 4.3 10 , which is added to the certainty factor above to bring the total for state 4 to 0.00328. Dervish would therefore track the new belief state to be 2, 4, assigning a very high likelihood to position 2 and a low likelihood to position 4. Empirically, Dervish's map representation and localization system have proven to be sufficient for navigation of four indoor office environments: the artificial office environment created explicitly for the 1994 National Conference on Artificial Intelligence; the psychology department, the history department and the computer science department at Stanford University. All of these experiments were run while providing Dervish with no notion of the distance between adjacent nodes in its topological map. It is a demonstration of the power of probabilistic localization that, in spite of the tremendous lack of action and encoder information, the robot is able to navigate several real-world office buildings successfully.

One open question remains with respect to Dervish's localization system. Dervish was not just a localizer but also a navigator. As with all multiple hypothesis systems, one must ask the question, how does the robot decide how to move, given that it has multiple possible robot positions in its representation? The technique employed by Dervish is a most common technique in the AMRics field: plan the robot's actions by assuming that the robot's actual position is its most likely node in the belief state. Generally, the most likely position is a good measure of the robot's actual world position. However, this technique has shortcomings when the highest and second highest most likely positions have similar values. In the case of Dervish, it nonetheless goes with the highest likelihood position at all times, save at one critical juncture. The robot's goal is to enter a target room and remain there. Therefore, from the point of view of its goal, it is critical that it finish navigating only when the robot has strong confidence in being at the correct final location. In this particular case, Dervish's execution module refuses to enter a room if the gap between the most likely position and the second likeliest position is below a preset threshold. In such a case, Dervish will actively plan a path that causes it to move further down the hallway in an attempt to collect more sensor data and thereby increase the relative likelihood of one position in the belief state. Although computationally unattractive, one can go further, imagining a planning system for robots such as Dervish for which one specifies a goal belief state rather than a goal position. The robot can then reason and plan in order to achieve a goal confidence level, thus explicitly taking into account not only robot position but also the measured likelihood of each position. An example of just such a procedure is the Sensory Uncertainty Field of Latombe [90], in which the robot must find a trajectory that reaches its goal while maximizing its localization confidence enroute.

2.6.3 Kalman Filter Localisation

The Markov localization model can represent any probability density function over robot position. This approach is very general but, due to its generality, inefficient. A successful alternative is to use a more compact representation of a specific class of probability densities. The Kalman filter does just this, and is an optimal recursive data processing algorithm. It incorporates all information,

regardless of precision, to estimate the current value of the variable of interest. A comprehensive introduction can be found in [46] and a more detailed treatment is presented in [28]. Figure 5.26 depicts the a general scheme of Kalman filter estimation, where the system has a control signal and system error sources as inputs. A measuring device enables measuring some system states with errors. The Kalman filter is a mathematical mechanism for producing an optimal estimate of the system state based on the knowledge of the system and the measuring device, the description of the system noise and measurement errors and the uncertainty in the dynamics models. Thus the Kalman filter fuses sensor signals and system knowledge in an optimal way. Optimality depends on the criteria chosen to evaluate the performance and on the assumptions. Within the Kalman filter theory the system is assumed to be linear and white with Gaussian noise. As we have discussed earlier, the assumption of Gaussian error is invalid for our AMR applications but, nevertheless, the results are extremely useful. In other engineering disciplines, the Gaussian error assumption has in some cases been shown to be quite accurate [46]. We begin with a subsection that introduces Kalman filter theory, then we present an application of that theory to the problem of AMR localization. Finally, the third subsection will present a case study of a AMR that navigates indoor spaces by virtue of Kalman filter localization.

A Gentle Introduction to Kalman Filter Theory

The **Kalman filter** method allows multiple measurements to be incorporated optimally into a single estimate of state. In demonstrating this, first we make the simplifying assumption that the state does **NOT** change²⁰ between the acquisition of the first and second measurement.

²⁰i.e., we are assuming the robot is stationary.

After presenting this static case, we can introduce dynamic prediction readily.

Static Estimation Let us assume we have taken two (2) measurements:

- one with an ultrasonic range sensor at time k , and
- one with a more precise laser range sensor at time $k + 1$.

Based on each measurement we are able to estimate the robot's position.

Such an estimate derived from the first sensor measurements is q_1 and the estimate of position based on the second measurement is q_2 .

As we know each measurement can be inaccurate, we wish to **modulate** these position estimates based on the **expected measurement error** from each sensor. Suppose we use two (2) variances (σ_1^2, σ_2^2) to predict the error associated with each measurement. We will assume a unimodal error distribution throughout the remainder of the Kalman filter approach, which gives us the two (2)

robot position estimates:

$$\hat{q}_1 = q_1 \quad \text{with variance} \quad \sigma_1^2, \quad (2.10)$$

$$\hat{q}_2 = q_2 \quad \text{with variance} \quad \sigma_2^2. \quad (2.11)$$

$$(2.12)$$

So now we have two (2) measurements available to estimate the robots position. The question we now have to answer is

How do we fuse these data to get the best estimate \hat{q} for the robot position?

We are assuming that there was no robot motion between time k and time $k + 1$, and therefore we can directly apply the same weighted least square technique:

$$S = \sum_{i=1}^n w_i (\hat{q} - q_i)^2 \quad (2.13)$$

with w being the weight of measurement i . To find the minimum error we set the derivative i of S equal to zero. which gives us:

$$\hat{q} = \frac{\sum_{i=1}^n w_i q_i}{\sum_{i=1}^n w_i} \quad (2.14)$$

Dynamic Estimation Following our previous model, we will now consider a robot which moves between successive sensor measurements. Suppose that the motion of the robot between times k and $k + 1$ is described by the velocity u and the noise w which represents the uncertainty of the actual velocity:

$$\frac{dx}{dt} = u + w \quad (2.15)$$

If we now start at time k , knowing the variance σ_k^2 of the robot position at this time and knowing the variance σ_w^2 of the motion, we obtain for the time k' just when the measurement is taken:

$$\hat{x}_{k'} = \hat{x}_k + u(t_{k+1} - t_k) \quad (2.16)$$

$$\hat{x}_{k'} = \hat{x}_k + u(t_{k+1} - t_k) \quad (2.17)$$

$$(2.18)$$

where ..

Kalman Filter Localisation

The Kalman filter is an optimal and efficient **sensor fusion** technique.

Application of the Kalman filter to localisation requires posing the robot localisation problem as a sensor fusion problem.

Recall that the basic probabilistic update of robot belief state can be segmented into two (2) phases:

- perception update, and
- action update

The fundamental difference between the Kalman filter approach and Markov localisation approach lies in the perception update process.

In Markov localisation, the entire perception²¹ is used to update each possible robot position in the belief state individually using Bayes formula.

²¹i.e., the robot's set of instantaneous sensor measurements.

In some cases, the perception is abstract, having been produced by a feature extraction mechanism.²² In other cases, as with Rhino, the perception consists of raw sensor readings.

²²as in Dervish.

By contrast, perception update using a Kalman filter is a **multi-step** process. The robot's total sensory input is treated, not as a monolithic whole, but as a set of extracted features which each relate to objects in the environment. Given a set of possible features, the Kalman filter is used to fuse the distance estimate from each feature to a matching object in the map. Instead of carrying out this matching process for many possible robot locations individually as in the Markov approach, the Kalman filter accomplishes the same probabilistic update by treating the whole, unimodal and Gaussian belief state at once. **Fig. 2.10** depicts the particular schematic for Kalman filter localisation.

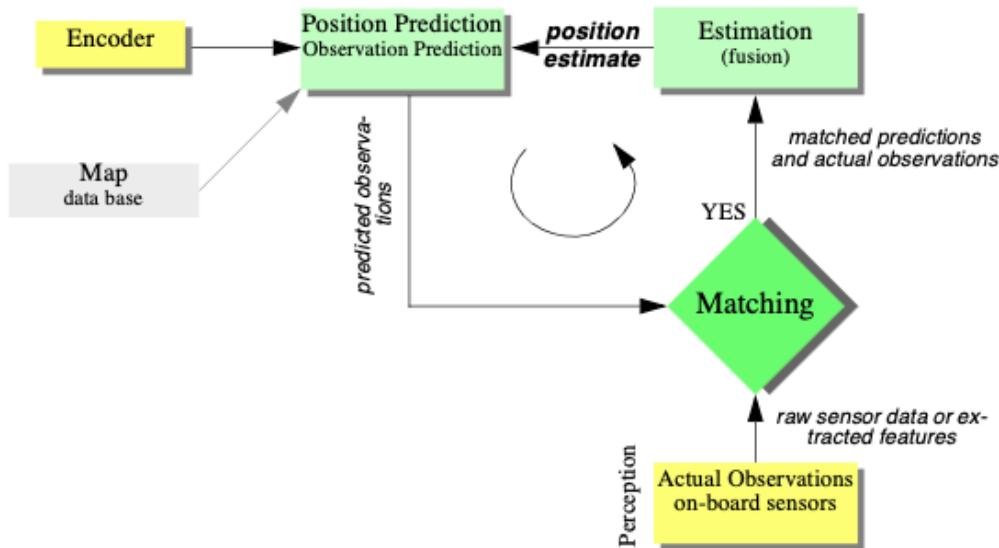


Figure 2.10: The schematic for the Kalman filter localisation

The first step is action update or position prediction, the straightforward application of a Gaussian error motion model to the robot's measured encoder travel. The robot then collects actual sensor data and extracts appropriate features²³ in the observation step. At the same time, based on its predicted position in the map, the robot generates a measurement prediction which identifies the features which the robot expects to find and the positions of those features. In matching the robot identifies the best pairings between the features actually extracted during observation and the

²³e.g. lines, doors, or even the value of a specific sensor

expected features due to measurement prediction. Finally, the Kalman filter can fuse the information provided by all of these matches in order to update the robot belief state in estimation.

2.7 Other Examples of Localisation Methods

Markov localisation and Kalman filter localisation have been two extremely popular strategies for research AMR systems navigating indoor environments. They have strong formal bases and therefore well-defined behavior. But there are a large number of other localisation techniques that have been used with varying degrees of success on commercial and research AMR platforms. We will not explore the space of all localisation systems in detail. Refer to surveys such as [4] for such information. There are, however, several categories of localisation techniques that deserve mention. Not surprisingly, many implementations of these techniques in commercial robotics employ modifications of the robot's environment, something that the Markov localisation and Kalman filter localisation communities eschew. In the following sections, we briefly identify the general strategy incorporated by each category and reference example systems, including as appropriate those that modify the environment and those that function without environmental modification.

2.7.1 Landmark-based Navigation

Landmarks are generally defined as **passive objects** in the environment which provide a high degree of localisation accuracy when they are within the robot's field of view. Mobile robots that make use of landmarks for localisation generally use artificial markers that have been placed by the robot's designers to make localisation easy.

The control system for a landmark-based navigator consists of two (2) discrete phases.

- When a landmark is in view, the robot localizes frequently and accurately, using action update and perception update to **track its position without cumulative error**.
- when the robot is in no landmark "zone", then only action update occurs, and the robot accumulates position uncertainty until the next landmark enters the robot's field of view.

The robot is thus effectively dead-reckoning from landmark zone to landmark zone. This in turn means the robot must consult its map carefully, ensuring that each motion between landmarks is sufficiently short, given its motion model, that it will be able to localize successfully upon reaching the next landmark.

Fig. 2.11 shows one instantiating of landmark-based localisation. The particular shape of the landmarks enables reliable and accurate pose estimation by the robot, which must travel using dead reckoning between the landmarks.

One key advantage of the landmark-based navigation approach is that a strong formal theory has been developed for this general system architecture [113]. In this work, the authors have shown precise assumptions and conditions which, when satisfied, guarantee that the robot will always be able to localize successfully. This work also led to a real-world demonstration of landmark-based localisation. Standard sheets of paper were placed on the ceiling of the Robotics Laboratory at



Figure 2.11: An illustration showing the object-level landmarks in blue-boxes. (a,b) shows two different indoor scenarios. The blue boxes represent the 3D object detection of object-level landmarks. The red dots indicate the nodes of the topological map. The yellow lines indicate the edges of the topological map. The green curve is the feasible navigation trajectory generated based on the proposed method [41].

Stanford University, each with a unique checkerboard pattern. A Nomadics 200 AMR was fitted with a monochrome CCD camera aimed vertically up at the ceiling. By recognizing the paper landmarks, which were placed approximately 2 meters apart, the robot was able to localize to within several centimeters, then move using dead-reckoning to another landmark zone.

The primary disadvantage of landmark-based navigation is that in general **it requires significant environmental modification**. Landmarks are local, and therefore a large number is usually required to cover a large factory area or research laboratory. For example, the Robotics Laboratory at Stanford made use of approximately 30 discrete landmarks, all affixed individually to the ceiling.

2.7.2 Globally Unique Localisation

The landmark-based navigation approach makes a strong general assumption:

when the landmark is in the robot's field of view, localisation is essentially perfect.

One way to reach the near perfect AMR localisation is to effectively enable such an assumption to be valid wherever the robot is located. It would be revolutionary if the robot's sensors immediately identified its particular location, uniquely, and repeatedly.

Such a strategy for localisation is surely aggressive, but the question of whether it can be done is primarily a question of sensor technology software. Clearly, such a localisation system would need to use a sensor which collects a very large amount of information.

Since vision does indeed collect far more information than other sensors, it has been used as the sensor of choice in research towards globally unique localisation.

If humans were able to look at an individual picture and identify the robot's location in a well-known environment, then one could argue that the information for globally unique localisation does exist within the picture. It must simply be interpreted correctly.

One such approach has been attempted by several researchers and involves constructing one or more image histograms to represent the information content of an image stably (see for example Figure 4.51 and Section 4.3.2.2). A robot using such an image histogramming system has been shown to uniquely identify individual rooms in an office building as well as individual sidewalks in an outdoor environment. However, such a system is highly sensitive to external illumination and provides only a level of localisation resolution equal to the visual footprint of the camera optics.

The Angular histogram depicted in Figure 5.37 is another example in which the robot's sensor values are transformed into an identifier of location. However, due to the limited information content of sonar ranging strikes, it is likely that two places in the robot's environment may have angular histograms that are too similar to be differentiated successfully.

One way of attempting to gather sufficient sonar information for global localisation is to allow the robot time to gather a large amount of sonar data into a local evidence grid (i.e. occupancy grid) first, then match the local evidence grid with a global metric map of the environment. In [115] the researchers demonstrate such a system as able to localize on-thefly even as significant changes are made to the environment, degrading the fidelity of the map. Most interesting is that the local evidence grid represents information well enough that it can be used to correct and update the map over time, thereby leading to a localisation system that provides corrective feedback to the environment representation directly. This is similar in spirit to the idea of taking rejected observed features in the Kalman filter localisation algorithm and using them to create new features in the map.

A most promising, new method for globally unique localisation is called Mosaic-based localisation [114]. This fascinating approach takes advantage of an environmental feature that is rarely used by AMRs: fine-grained floor texture. This method succeeds primarily because of the recent ubiquity of very fast processors, very fast cameras and very large storage media.

The robot is fitted with a high-quality high-speed CCD camera pointed toward the floor, ideally situated between the robot's wheels and illuminated by a specialized light pattern off the camera axis to enhance floor texture. The robot begins by collecting images of the entire floor in the robot's workspace using this camera. Of course the memory requirements are significant, requiring a 10GB drive in order to store the complete image library of a 300×300 meter area. Once the complete image mosaic is stored, the robot can travel any trajectory on the floor while tracking its own position without difficulty. Localisation is performed by simply recording one image, performing action update, then performing perception update by matching the image to the mosaic database using simple techniques based on image database matching. The resulting performance has been impressive: such a robot has been shown to localize repeatedly with 1mm precision while moving at 25 km/hr. The key advantage of globally unique localisation is that, when these systems function correctly, they greatly simplify robot navigation. The robot can move to any point and will always be assured

of localizing by collecting a sensor scan. But the main disadvantage of globally unique localisation is that it is likely that this method will never offer a complete solution to the localisation problem. There will always be cases where local sensory information is truly ambiguous and, therefore, globally unique localisation using only current sensor information is unlikely to succeed. Humans often have excellent local positioning systems, particularly in non-repeating and well-known environments such as their homes. However, there are a number of environments in which such immediate localisation is challenging even for humans: consider hedge mazes and large new office buildings with repeating halls that are identical. Indeed, the mosaic-based localisation prototype described above encountered such a problem in its first implementation. The floor of the factory floor had been freshly painted and was thus devoid of sufficient micro-fractures to generate texture for correlation. Their solution was to modify the environment after all, painting random texture onto the factory floor.

2.7.3 Positioning Beacon systems

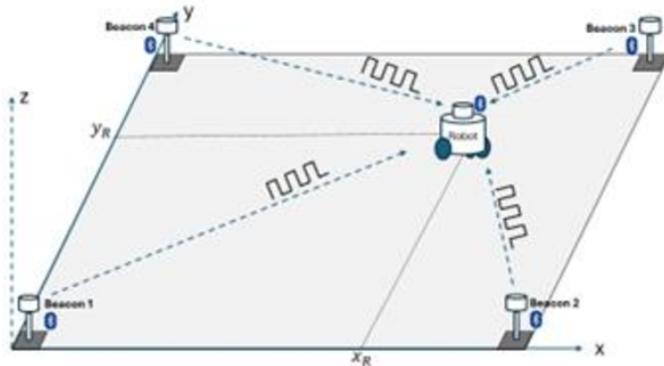


Figure 2.12

With most beacon systems, the design depicted depends foremost upon geometric principles to effect localisation. In this case the robots must know the positions of the two pinger units in the global coordinate frame in order to localize themselves to the global coordinate frame. A popular type of beacon system in industrial robotic applications is depicted in Figure 5.39. In this case beacons are retroreflective markers that can be easily detected by a AMR based on their reflection of energy back to the robot. Given known positions for the optical retroreflectors, a AMR can identify its position whenever it has three such beacons in sight simultaneously. Of course, a robot with encoders can localize over time as well, and does not need to measure its angle to all three beacons at the same instant. The advantage of such beacon-based systems is usually extremely high engineered reliability. By the same token, significant engineering usually surrounds the installation of such a system in a specific commercial setting. Therefore, moving the robot to a different factory floor will be both time-consuming and expensive. Usually, even changing the routes used by the robot will require serious re-engineering.

2.7.4 Route-Based Localisation

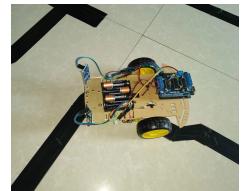
Even more reliable than beacon-based systems are route-based localisation strategies. In this case, the route of the robot is explicitly marked so that it can determine its position, not relative to some global coordinate frame, but relative to the specific path it is allowed to travel.²⁴ There are many techniques for marking such a route and the subsequent intersections.

In all cases, one is effectively creating a railway system, except the railway system is somewhat more flexible and certainly more human-friendly actual rail.

For example, high UV-reflective, optically transparent paint can mark the route such that only the robot, using a specialized sensor, easily detects it. Alternatively, a guide wire buried underneath the hall can be detected using inductive coils located on the robot chassis.

In all such cases, the robot localisation problem is effectively trivialized by forcing the robot to always follow a prescribed path. While this may remove the **autonomous** part of AMR, there are industrial unmanned guided vehicles that do deviate briefly from their route in order to avoid obstacles. Nevertheless, the cost of this extreme reliability is obvious:

the robot is much more inflexible given such localisation means, and therefore any change to the robot's behavior requires significant engineering and time.



²⁴A perfect example for these kind of localisation is the traditional line following robot. The robot does not need to know where it is as its only job is to make sure the line it is following is within its vision [42].

2.8 Building Maps

Humans are excellent navigators due to their remarkable ability to build cognitive maps [43] which form the basis of spatial memory [44], [45]. However, when it comes to AMR, we unfortunately need to be more hands on.

All of the localisation strategies we have discussed previously require active human effort to install the robot into a space. Artificial environmental modifications may be necessary to reduce ambiguity [46]. Even if this is not so, a map of the environment must be created for the robot.

But a robot which localizes successfully has the right sensors for detecting the environment, and so the robot ought to build its own map.

This ambition goes to the heart of AMR. In prose, we can express our eventual goal as follows:

Starting from an arbitrary initial point, a AMR should be able to autonomously explore the environment with its on-board sensors, gain knowledge about it, interpret the scene, build an appropriate map and localize itself relative to this map.

While we have system which allows certain level of intelligence to robots, most applications require a connected network or a central node to achieve any autonomous action [47], [48]. Accomplishing this goal purely using internal components in a robust is probably years away, but an important sub-goal is the invention of techniques for autonomous creation and modification of an environment map. Of course a AMR's sensors have only limited range, and so it must physically explore its environment to build such a map. So, the robot must not only create a map but it must do so while moving and localizing to explore the environment. This is often called the Simultaneous Localisation and Mapping (SLAM) problem,²⁵ arguably the most difficult problem specific to AMR systems.

²⁵Computational problem of constructing or updating a map of an unknown environment while simultaneously keeping track of an agent's location within it. While this initially appears to be a chicken or the egg problem, there are several algorithms known to solve it in, at least approximately and in reasonable time for certain environments. Popular solutions include the particle filter, extended Kalman filter, covariance intersection, and GraphSLAM. SLAM algorithms are based on concepts in computational geometry and computer vision, and are used in robot navigation, robotic mapping and odometry for virtual reality or augmented reality.



Figure 2.13: 2005 DARPA Grand Challenge winner Stanley performed SLAM as part of its autonomous driving system [49].

The reason why SLAM is difficult is born precisely from the interaction between the robot's position updates as it localises and its mapping actions. If a AMR updates its position based on an observation of an imprecisely known feature, the resulting position estimate becomes correlated with the feature

location estimate. Similarly, the map becomes correlated with the position estimate if an observation taken from an imprecisely known position is used to update or add a feature to the map.

For localisation the robot needs to know where the features are whereas for map building the robot needs to know where it is on the map.

The only path to a complete and optimal solution to this joint problem is to consider all the correlations between position estimation and feature location estimation. Such cross-correlated maps are called stochastic maps [50]. Unfortunately, implementing such an optimal solution is computationally prohibitive.

2.8.1 Stochastic Map Technique

Fig. 2.14 shows a general schematic incorporating map building and maintenance into the standard localisation loop depicted by Figure (5.29) during discussion of Kalman filter localisation [9]. The added arcs represent the additional flow of information that occurs when there is an imperfect match between observations and measurement predictions.

Unexpected observations will affect the creation of new features in the map whereas unobserved measurement predictions will affect the removal of features from the map. As discussed earlier, each specific prediction or observation has an unknown exact value and so it is represented by a distribution. The uncertainties of all of these quantities must be considered throughout this process.

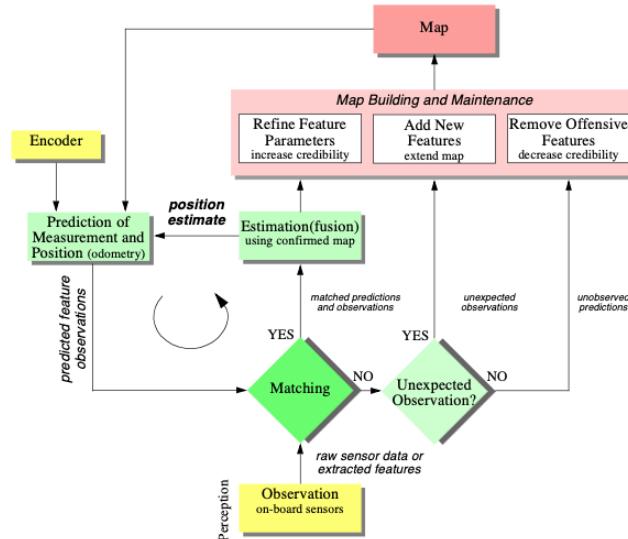


Figure 2.14: General schematic for concurrent localization and map building.

The new type of map we are creating not only has features in it as did previous maps, but it also has varying degrees of probability that each feature is indeed part of the environment.

We represent this new map M with a set n of probabilistic feature locations \hat{z}_t , each with the covariance matrix Σ_t and an associated **credibility factor** c_t between 0 and 1.

The purpose of c_t is to quantify the belief in the existence of the feature in the environment (see Fig. (5.41)):

$$M = \left\{ z_t, \Sigma_t, c_t \mid (1 \leq t \leq n) \right\} \quad (2.19)$$

In contrast to the map used for Kalman filter localisation previously, the map M is **NOT** assumed to be **precisely known** as it will be created by an uncertain robot over time. This is why the features \hat{z} are described with associated covariance matrices Σ_t .

Similar to Kalman filter localisation, the matching steps has three (3) outcomes in regard to measurement predictions and observations:

- matched prediction and observations,
- unexpected observations, and
- unobserved predictions

Localisation, or the position update of the robot, proceeds as before. However, the map is also updated now, using all three outcomes and complete propagation of all the correlated uncertainties.

The interesting concept in this modelling is the **credibility factor** c_t , which governs the likelihood that the mapped feature is indeed in the environment.

How should the robot's failure to match observed features to a particular map feature reduce that map feature's credibility?

How should the robot's success at matching a mapped feature increase the chance that the mapped feature is "correct?"

As an example, in [51] the following function is proposed for calculating credibility:

$$c_t(k) = 1 - \exp \left(- \left(\frac{n_s}{a} - \frac{n_u}{b} \right) \right) \quad (2.20)$$

where a and b define the **learning** and **forgetting** rate and n_s and n_u are the number of matched and unobserved predictions up to time k , respectively. The update of the covariance matrix Σ_t building the feature positions and the robot's position are strongly correlated.

This forces us to use a stochastic map, in which all cross-correlations must be updated in each cycle.

The stochastic map consists of a stacked system state vector:

$$\mathbf{x} = [x_r(k) \quad x_1(k) \quad x_2(k) \quad \dots \quad x_n(k)]^T \quad (2.21)$$

and a system state covariance matrix:

$$\Sigma = \begin{bmatrix} C_{rr} & C_{r1} & \cdots & C_{rn} \\ C_{1r} & C_{11} & \cdots & C_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ C_{nr} & C_{n1} & \cdots & C_{nn} \end{bmatrix} \quad (2.22)$$

where the index r stands for the robot and the index i = 1 to n for the features in the map.

In contrast to localization based on an a priori accurate map, in the case of a stochastic map the cross-correlations must be maintained and updated as the robot is performing automatic map-building. During each localization cycle, the cross-correlations robot-to-feature and feature-to-robot are also updated. In short, this optimal approach requires every value in the map to depend on every other value, and therein lies the reason that such a complete solution to the automatic mapping problem is beyond the reach of even todays computational resources.

2.8.2 Other Mapping Techniques

The AMR research community has spent significant research effort on the problem of automatic mapping, and has demonstrating working systems in many environments without having solved the complete stochastic map problem described earlier. This field of mobile robotics research is extremely large, and this text will not present a comprehensive survey of the field. Instead, we present below two key considerations associated with automatic mapping, together with brief discussions of the approaches taken by several automatic mapping solutions to overcome these challenges.

Cyclic Environments

Possibly the single hardest challenge for automatic mapping to be conquered is to correctly map cyclic environments. The problem is simple: given an environment that has one or more loops or cycles (e.g. four hallways that intersect to form a rectangle), create a globally consistent map for the whole environment. This problem is hard because of the fundamental behavior of automatic mapping systems: the maps they create are not perfect. And, given any local imperfection, accumulating such imperfections over time can lead to arbitrarily large global errors between a map, at the macro level, and the real world, as shown in Figure (5.42). Such global error is usually irrelevant for AMR localisation and navigation. After all, a warped map will still serve the robot perfectly well so long as the local error is bounded. However, an extremely large loop still eventually returns to the same spot, and the robot must be able to note this fact in its map. Therefore, global error does indeed matter in the case of cycles. In some of the earliest work attempting to solve the cyclic environment problem, [116] used a purely topological representation of the environment, reasoning that the topological representation only captures the most abstract, most important features and avoids a great deal of irrelevant detail. When the robot arrives at a topological node that could be the same as a previously visited and mapped node (e.g. similar distinguishing features), then

the robot postulates that it has indeed returned to the same node. To check this hypothesis, the robot explicitly plans and moves to adjacent nodes to see if its perceptual readings are consistent with the cycle hypothesis. With the recent popularity of metric maps such as fixed decomposition grid representations, the cycle detection strategy is not as straightforward. Two important features are found in most autonomous mapping systems that claim to solve the cycle detection problem. First, as with many recent systems, these mobile robots tend to accumulate recent perceptual history to create small-scale local sub-maps [117, 118, 119]. Each sub-map is treated as a single sensor during the robot's position update. The advantage of this approach is two-fold. Because odometry is relatively accurate over small distances, the relative registration of features and raw sensor strikes in a local sub-map will be quite accurate. In addition to this, the robot will have created a virtual sensor system with a significantly larger horizon than its actual sensor system's range. In a sense, this strategy at the very least defers the problem of very large cyclic environments by increasing the map scale that can be handled well by the robot. The second recent technique for dealing with cycle environments is in fact a return to the topological representation. Some recent automatic mapping systems will attempt to identify cycles by associating a topology with the set of metric sub-maps, explicitly identifying the

loops first at the topological level. In the case of [118] for example, the topological level loop is identified by a human who pushes a button at a known landmark position. In the case of [119] the topological level loop is determined by performing correspondence tests between sub-maps, postulating that two sub-maps represent the same place in the environment when the correspondence is good. One could certainly imagine other augmentations based on known topological methods. For example, the globally unique localisation methods described in Section (5.7) could be used to identify topological correctness. It is notable that the automatic mapping research of the present has, in many ways, returned to the basic topological correctness question that was at the heart of some of the earliest automatic mapping research in AMRics more than a decade ago. Of course, unlike that early work, today's automatic mapping results boast correct cycle detection combined with high-fidelity geometric maps of the environment.

Dynamic Environments

A second challenge extends **NOT** just to existing autonomous mapping solutions but even to the basic execution of the stochastic map approach.

All previously mentioned strategies tend to assume the environment is either unchanging or changes in ways that are virtually insignificant. Such assumptions are certainly valid with respect to some environments, such as for example the computer science department of a university at 3:00 AM.

However, for many practical applications, this assumption is lacking at best. In the case of wide-open spaces that are popular gathering places for humans, there is rapid change in the freespace and a vast majority of sensor strikes represent detection of the transient humans rather than fixed surfaces such as the perimeter wall. Another class of dynamic environments are spaces such as factory

floors and warehouses, where the objects being stored redefine the topology of the pathways on a day-to-day basis as shipments are moved in and out.

In all such dynamic environments, an automatic mapping system should capture the salient²⁶ objects detected by its sensors and, furthermore, the robot should have the flexibility to modify its map as the position of these salient objects changes.

²⁶In this context, salient means anything which is sticking out.

The subject of **continuous mapping**, or mapping of dynamic environments is to some degree a direct outgrowth of successful strategies for automatic mapping of unfamiliar environments.

For example, in the case of stochastic mapping using the credibility factor c_t mechanism, the credibility equation can continue to provide feedback regarding the probability of existence of various mapped features after the initial map creation process is ostensibly complete. Therefore, a mapping system can become a map-modifying system by simply continuing to operate. This is most effective, of course, if the mapping system is real-time and incremental. If map construction requires off-line global optimisation, then the desire to make small-grained, incremental adjustments to the map is more difficult to satisfy. Earlier we stated that a mapping system should capture only the salient objects detected by its sensors. One common argument for handling the detection of, for instance, humans in the environment is that mechanisms such as c serve to be mapped in the first place. For example, in [117] the authors develop a system based on a set of local occupancy grids²⁷ and a global occupancy grid.

²⁷called evidence grids

Each time the robot's most recent local evidence grid is used to update a region of the global occupancy grid, extraneous occupied cells in the global occupancy grid are freed if the local occupancy grid detected no objects (with high confidence) at those same positions. The general solution to the problem of detecting salient features, however, begs a solution to the perception problem in general. When a robot's sensor system can reliably detect the difference between a wall and a human, using for example a vision system, then the problem of mapping in dynamic environments will become significantly more straightforward. We have discussed just two important considerations for automatic mapping. There is still a great deal of research activity focusing on the general map building and localisation problem [9, 6, 47, 48, 49, 50, 75, 77]. However, there are few groups working on the general problem of probabilistic map building (i.e. stochastic maps) and, so far, a consistent and absolutely general solution has yet to be found. This field is certain to produce significant new results in the next several years, and as the perceptual power of robots improves we expect the payoff to be greatest here.

Bibliography

- [1] Farnell. *Rotary Encoder, Module, Optical, Incremental, 500 PPR, 0 Detents, Vertical, Without Push Switch*. 2025. URL: <https://at.farnell.com/en-AT/broadcom-limited/aedb-9140-a13/encoder-3channel-500cpr-8mm/dp/1161087>.
- [2] Flyrobo. *GY-26 Digital Electronic Compass Sensor Module*. 2025. URL: <https://www.flyrobo.in/gy-26-digital-electronic-compass-sensor-module>.
- [3] FindLight. *Optical Gyroscopes: Measuring Rotational Changes With Sagnac Effect*. 2025. URL: <https://www.findlight.net/blog/optical-gyroscopes-measuring-rotations/>.
- [4] PiHut. *HC-SR04 Ultrasonic Range Sensor on the Raspberry Pi*. 2025. URL: <https://thepihut.com/blogs/raspberry-pi-tutorials/hc-sr04-ultrasonic-range-sensor-on-the-raspberry-pi>.
- [5] Reinhold. *Structured light sources on display at the 2014 Machine Vision Show in Boston*. 2014. URL: https://commons.wikimedia.org/wiki/File:Structured_light_sources.agr.jpg.
- [6] Andrzej. *CCD image sensor SONY ICX493AQA 10,14 (Gross 10,75) M pixels APS-C 1.8" 28.328mm (23.4 x 15.6 mm) from module IS-026 from digital camera SONY DSLR-A200 or DSLR-A300 sensor side*. 2014. URL: https://commons.wikimedia.org/wiki/File:CCD_SONY_ICX493AQA_sensor_side.jpg.
- [7] TeledyneCCD. *How a Charge Coupled Device (CCD) Image Sensor Works*. 2020. URL: https://www.teledyneimaging.com/media/1300/2020-01-22_e2v_how-a-charge-coupled-device-works_web.pdf.
- [8] K. Hirakawa and T.W. Parks. "Chromatic adaptation and white-balance problem". In: *IEEE International Conference on Image Processing 2005*. Vol. 3. 2005, pp. III–984. DOI: [10.1109/ICIP.2005.1530559](https://doi.org/10.1109/ICIP.2005.1530559).
- [9] Fstoppers. *Is There a Difference Between Color Temperature and White Balance?* 2022. URL: <https://fstoppers.com/natural-light/there-difference-between-color-temperature-and-white-balance-596031>.
- [10] Adrian Ilie and Greg Welch. "Ensuring color consistency across multiple cameras". In: *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*. Vol. 2. IEEE. 2005, pp. 1268–1275.
- [11] Teledyne. *Saturation and Blooming*. 2025. URL: <https://www.photometrics.com/learn/imaging-topics/saturation-and-blooming>.

- [12] Zhen Zhang et al. "Analysis and simulation to excessive saturation effect of CCD". In: *2nd International Symposium on Laser Interaction with Matter (LIMIS 2012)*. Vol. 8796. SPIE. 2013, pp. 96–102.
- [13] Baumer. *Operating principle and features of CMOS sensors*. 2025. URL: <https://www.baumer.com/int/en/service-support/function-principle/operating-principle-and-features-of-cmos-sensors/a/EMVA1288>.
- [14] econ Systems. *CMOS camera module*. <https://www.directindustry.com/prod/e-con-systems/product-168594-2365044.html>. URL: <https://www.directindustry.com/prod/e-con-systems/product-168594-2365044.html>.
- [15] TS Holst and GC Lomheim. *CMOS/CCD Sensors*. JCD publishing, 2007.
- [16] Mdf. *A photon noise simulation*. 2010. URL: <https://commons.wikimedia.org/wiki/File:Photon-noise.jpg>.
- [17] Mark-j. *Open Camera Blog*. 2018. URL: <https://sourceforge.net/p/opencamera/blog/2018/09/focus-bracketing-with-open-camera/>.
- [18] UoW. *How Mars rovers use artificial intelligence*. 2023. URL: <https://online.wlv.ac.uk/how-mars-rovers-use-artificial-intelligence/>.
- [19] Tim Bailey. "Mobile robot localisation and mapping in extensive outdoor environments". PhD thesis. Citeseer, 2002.
- [20] Sean Campbell et al. "Where am I? Localization techniques for mobile robots a review". In: *2020 6th International Conference on Mechatronics and Robotics Engineering (ICMRE)*. IEEE. 2020, pp. 43–47.
- [21] Victoria J Hodge. "Sensors and data in mobile robotics for localisation". In: *Encyclopedia of Data Science and Machine Learning* (2023), pp. 2223–2238.
- [22] Omar Jaradat et al. "Challenges of safety assurance for industry 4.0". In: *2017 13th European Dependable Computing Conference (EDCC)*. IEEE. 2017, pp. 103–106.
- [23] David Filliat and Jean-Arcady Meyer. "Map-based navigation in mobile robots:: I. a review of localization strategies". In: *Cognitive systems research* 4.4 (2003), pp. 243–282.
- [24] John J Leonard and Hugh F Durrant-Whyte. *Directed sonar sensing for mobile robot navigation*. Vol. 175. Springer Science & Business Media, 2012.
- [25] Michael G Wing, Aaron Eklund, and Loren D Kellogg. "Consumer-grade global positioning system (GPS) accuracy and reliability". In: *Journal of forestry* 103.4 (2005), pp. 169–173.
- [26] Nicola Bellotto and Huosheng Hu. "People tracking and identification with a mobile robot". In: *2007 International Conference on Mechatronics and Automation*. IEEE. 2007, pp. 3565–3570.
- [27] Mohan Sridharan and Peter Stone. "Color learning and illumination invariance on mobile robots: A survey". In: *Robotics and Autonomous Systems* 57.6-7 (2009), pp. 629–644.
- [28] Diego Galar and Uday Kumar. "Chapter 1 - Sensors and Data Acquisition". In: *eMaintenance*. Ed. by Diego Galar and Uday Kumar. Academic Press, 2017, pp. 1–72. ISBN: 978-0-12-811153-6. DOI: <https://doi.org/10.1016/B978-0-12-811153-6.00001-4>. URL: <https://www.sciencedirect.com/science/article/pii/B9780128111536000014>.

- [29] David R Williams. "Aliasing in human foveal vision". In: *Vision research* 25.2 (1985), pp. 195–205.
- [30] maksim. *An example of a poorly sampled brick pattern*. 2006. URL: https://commons.wikimedia.org/wiki/File:Moire_pattern_of_bricks_small.jpg.
- [31] Gaddi Blumrosen, Ben Fishman, and Yossi Yovel. "Noncontact wideband sonar for human activity detection and classification". In: *IEEE Sensors Journal* 14.11 (2014), pp. 4043–4054.
- [32] Angelo M Sabatini and Valentina Colla. "A method for sonar based recognition of walking people". In: *Robotics and Autonomous Systems* 25.1-2 (1998), pp. 117–126.
- [33] Parag H Batavia and Illah Nourbakhsh. "Path planning for the Cye personal robot". In: *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000) (Cat. No. 00CH37113)*. Vol. 1. IEEE. 2000, pp. 15–20.
- [34] J Reuter. "Scan-and featurebased multiple hypothesis tracking for mobile robot localization: a data fusion approach". In: *IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No. 99CH37028)*. Vol. 4. IEEE. 1999, pp. 714–719.
- [35] Péter Fankhauser, Michael Bloesch, and Marco Hutter. "Probabilistic terrain mapping for mobile robots with uncertain localization". In: *IEEE Robotics and Automation Letters* 3.4 (2018), pp. 3019–3026.
- [36] Yonhap News Agency. *Museum guide robot*. 2018. URL: <https://en.yna.co.kr/view/PYH20181221009400341>.
- [37] Oliver Brock and Oussama Khatib. "High-speed navigation using the global dynamic window approach". In: *Proceedings 1999 ieee international conference on robotics and automation (Cat. No. 99CH36288C)*. Vol. 1. IEEE. 1999, pp. 341–346.
- [38] J Borenstein and Y Koren. "Fast obstacle avoidance for mobile robots". In: *IEEE Trans Rob Autom.* v7 (), pp. 278–287.
- [39] Rodney Brooks. "A robust layered control system for a mobile robot". In: *IEEE journal on robotics and automation* 2.1 (1986), pp. 14–23.
- [40] Illah Nourbakhsh, Rob Powers, and Stan Birchfield. "DERVISH an office-navigating robot". In: *AI magazine* 16.2 (1995), pp. 53–53.
- [41] Fan Wang et al. "Object-based reliable visual navigation for mobile robot". In: *Sensors* 22.6 (2022), p. 2387.
- [42] AaaravG. *How to Make Line Follower Robot Using Arduino*. 2018. URL: <https://www.instructables.com/Line-Follower-Robot-Using-Arduino-2/>.
- [43] Timothy P McNamara, James K Hardy, and Stephen C Hirtle. "Subjective hierarchies in spatial memory." In: *Journal of Experimental Psychology: Learning, Memory, and Cognition* 15.2 (1989), p. 211.
- [44] Marvin M Chun and Yuhong Jiang. "Contextual cueing: Implicit learning and memory of visual context guides spatial attention". In: *Cognitive psychology* 36.1 (1998), pp. 28–71.

- [45] Chenguang Huang et al. "Visual language maps for robot navigation". In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2023, pp. 10608–10615.
- [46] Daniel Meyer-Delius et al. "Using artificial landmarks to reduce the ambiguity in the environment of a mobile robot". In: *2011 IEEE International Conference on Robotics and Automation*. IEEE. 2011, pp. 5173–5178.
- [47] Jakub Hazik et al. "Fleet management system for an industry environment". In: *Journal of Robotics and Control (JRC)* 3.6 (2022), pp. 779–789.
- [48] Elias Xidias, Paraskevi Zacharia, and Andreas Nearchou. "Intelligent fleet management of autonomous vehicles for city logistics". In: *Applied Intelligence* 52.15 (2022), pp. 18030–18048.
- [49] Kivaan. *An official DARPA photograph of Stanley at the 2005 DARPA Grand Challenge*. 2007. URL: <https://commons.wikimedia.org/wiki/File:Stanley2.JPG>.
- [50] Shuji Hashimoto et al. "Humanoid robots in waseda universityhadaly-2 and wabian". In: *Autonomous Robots* 12 (2002), pp. 25–38.
- [51] Raymond J Carroll and David Ruppert. *Transformation and weighting in regression*. Chapman and Hall/CRC, 2017.

