Mathematic model of an IMU

What is an IMU?

An IMU is an Inertial Measurement Unit, generally being either 6-DoF or 9-DoF.

An 6-DoF IMU is made up of 3 gyroscopes and 3 accelerometers (one in each of X, Y, Z axes).

A 9-DoF IMU also includes a 3-axis magnetometer.

Gyroscopes measure angular velocity across an axis. Gyroscopes are not affected by external forces and acceleration meaning they work well under dynamic conditions when rotational velocities are high, however they drift significantly with regard to time. Thus the simplest filtering operation on gyroscope data is a high pass filter to remove low frequency drift.

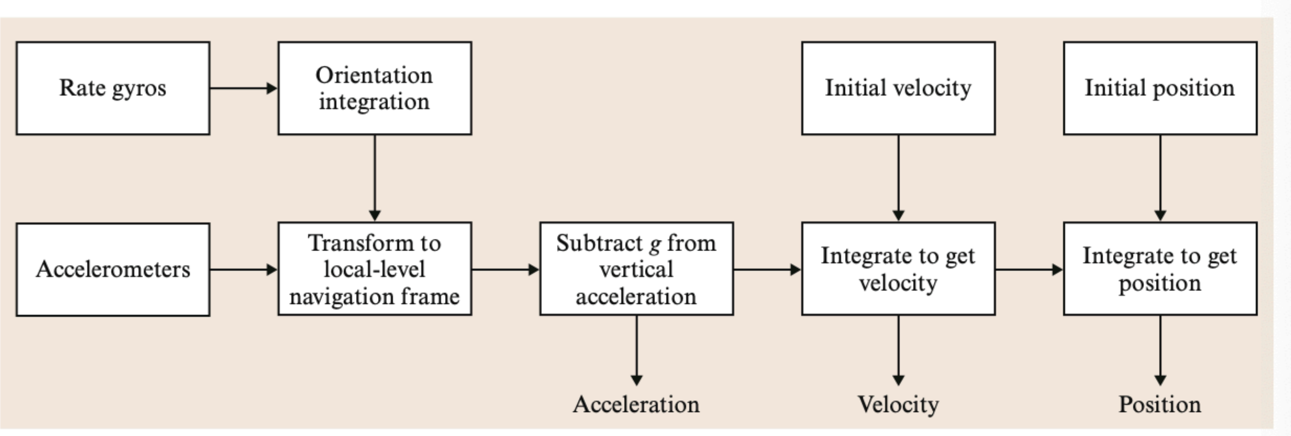
Accelerometers measure effective acceleration along an axis. Accelerometers are affected by vibration and other external forces and hence cannot be directly used for computing angles / attitude accurately. However, accelerometers work well in static conditions as opposed to gyroscopes. Hence, the simplest filtering operation performed on accelerometer data is a low pass filter to remove dynamic noise from vibrations and other external factors.

Magnetometers measure the Earth's magnetic field which can be used to obtain orientation information as well. For indoor operation with lots of metal structure, magnetometers are generally inaccurate and excluded from data fusion. Magnetometers are usually called Digital Compasses as they can be directly used to compute the North Pole direction.

The sensing technology of an IMU is typically integrated with an on-board computation unit to produce an attitude and heading reference unit (AHRS), a single device which maintains a 6-DOF estimate of the pose of the vehicle $[x, y, z, \mathrm{roll}, \mathrm{pitch}, \mathrm{yaw}]$

Distinction between an IMU and AHRS is often unclear.

The basic computational task of an IMU is shown below:



Gyroscope data is integrated to maintain an ongoing estimate of vehicle orientation θ. Accelerometer data is transformed via the current estimate of the vehicle orientation relative to gravity, so that the gravity vector can be estimated and extracted from the measurement. The resulting acceleration is then integrated to obtain vehicle velocity and then integrated again to obtain position.

IMU's are extremely sensitive to measurement errors in the underlying gyroscopes and accelerometers.

Drift in the gyroscopes leads to mis-estimates of the vehicle orientation relative to gravity, resulting in an incorrect cancellation of the gravity vector.

As the accelerometer data is integrated twice, any residual gravity vector will result in a quadratic error in position

As it is never possible to completely eliminate the gravity vector and this and any other error is integrated over time drift is a fundamental issue for any IMU.

Given a sufficiently long period of operation all IMU's eventually drift and reference to some external measurement is required to correct this. For many field robots GPS has become an effective source for these external correction

Gyroscope mathematical model

Gyroscopic systems aim to measure changes in vehicle orientation. Rotating frames are not inertial frame and thus many physical systems appear to behave in a non-Newtonian manner if the frame is rotating. By measuring these deviations from what would be expected in a Newtonian frame the underlying self-rotation can be determined.

The mathematical model for a gyroscope is:

$$\omega = \hat{\omega} + \mathbf{b}_g \ + \mathbf{n}_g$$

Where:

- ω is the measured angular velocity from the gyroscope

- $\hat{\omega}$ is the latent ideal angular velocity we wish to recover
- $\mathbf{b}_g$ is the gyroscope bias which changes with time and other factors such as temperature
- $\mathbf{n}_g$ is the white gaussian gyroscope noise

The gyroscope bias is modelled as:

$$\dot{\mathbf{b}}_g = \mathbf{b}_{bg}(t) \sim \mathcal{N}(0, Q_g)$$

Where

- $\mathbf{b}_{bg}(t)$ is the bias random walk. i.e. the bias is modelled as a random walk process, evolving randomly over time
- Q_g is the covariance matrix which models gyroscope noise [Multivariate normal distribution > Covariance matrix]

Q_g quantifies the magnitude and relationships of the noise in the gyroscope bias

$$Q_g = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_y^2 & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_z^2 \end{bmatrix}$$

The diagonal entries represent the variance of the noise in each axis and the off-diagonal entries represent the covariance between axis (how noise in one axis is related to noise in another)

Thus, the bias model suggests that at any given time, the gyroscope bias is a random value that evolves continuously over time. This evolution is unpredictable but statistically described by a multivariate Gaussian distribution with a mean of 0 (the bias changes are centered around zero, so on average, there’s no systematic trend in any specific direction) and a covariance of Q_g (determines how much the bias can vary and how correlated the changes in different axes are)

$\mathbf{n}_g$ is a random noise term added to the gyroscope's output, representing measurement inaccuracies due to imperfections in the sensor. The term "white" refers to the fact that the value of this noise is independent of its value at any other time. There’s no predictable trend or relationship in the noise over time.

But we still model this using a Gaussian distribution:

$$\mathbf{n}_g \sim \mathcal{N}(0, R_g)$$

Where:

- R_g specifies the variance of the noise along each axis and the correlation (if any) between the noise on different axes.

The white Gaussian noise in gyroscopes has the following characteristics:

- Zero mean: $\mathbb{E}[\mathbf{n}_g] = 0$, meaning the noise doesn't introduce a systematic offset
- Constant variance over time

Accelerometer mathematical model

Accelerometers measure external forces acting on vehicles. They are sensitive to all external forces acting on them including gravity. Different mechanisms can be used to transduce external forces into a computer readable signal.

Mechanical accelerometers are spring-mass-damper systems with some mechanism for external monitoring.

Piezoelectrical accelerometers use the property of certain crystals to generate a voltage across them when stressed. A small mass can be positioned so that it is only supported by the crystal, and as forces cause the mass to act upon the crystal this induces a voltage that can be measured

The mathematical model for an accelerometer is:

$$\mathbf{a} = R^T(\hat{\mathbf{a}} - \mathbf{g}) + \mathbf{b}_a + \mathbf{n}_a$$

Where:

- $\mathbf{a}$ is the measured acceleration from the accelerometer
- $\hat{\mathbf{a}}$ is the latent ideal acceleration we wish to recover
- R is the orientation of the sensor in the world frame, i.e. the rotation matrix that describes the sensor orientation relative to the world frame
- R^T is the rotation matrix that converts a vector from the world frame into the sensor's frame
- $\mathbf{g}$ is the acceleration due to gravity in the world frame
- $\mathbf{b}_a$ is the accelerometer bias which changes with time and other factors such as temperature
- $\mathbf{n}_a$ is the white gaussian accelerometer noise

Thus the model is the latent acceleration is subtracted from gravity and then transformed into the sensor's frame and bias and noise are added.

The accelerometer bias is modelled as:

$$\dot{\mathbf{b}}_a = \mathbf{b}_{ba}(t) \sim \mathcal{N}(0, Q_a)$$

Where:

- $\mathbf{b}_{ba}(t)$ is the bias random walk. i.e. the bias is modelled as a random walk process, evolving randomly over time
- Q_a is the covariance matrix which models accelerometer noise

Here the orientation of the sensor is either known from external sources such as estimated by sensor fusion.

Magnetometer mathematical model

The mathematical model for a magnetometer is:

$$\mathbf{m} = R^T(\hat{\mathbf{m}} + \mathbf{b}_m + \mathbf{n}_m)$$

Where

- $\mathbf{m}$ is the measured magnetic field vector in the sensor frame
- $\hat{\mathbf{m}}$ is the latent ideal magnetic field vector in the Earth frame (the Earth's magnetic field at the sensor's location)
- R is the rotation matrix that describes the sensor's orientation relative to the world frame
- R^T is the transpose of the rotation matrix, transforming the world-frame magnetic field into the sensor's frame
- $\mathbf{b}_m$ is the magnetometer bias, which includes hard-iron distortions (constant offsets due to nearby ferromagnetic materials) and soft-iron distortions (scaling and misalignment due to nearby materials affecting the field)
- $\mathbf{n}_m$ is white Gaussian noise representing random errors in the sensor

The magnetometer bias evolves over time as a random process, similar to the accelerometer and gyroscope:

$$\dot{\mathbf{b}}_m = \mathbf{b}_{bm}(t) \sim \mathcal{N}(0, Q_m)$$

where

- $\mathbf{b}_{bm}(t)$ is a random walk process modelling bias drift over time
- Q_m is the covariance matrix modelling noise in the bias

A more realistic model separates the bias model into hard-iron and soft-iron distortions:

$$\mathbf{m} = S_m R^T(\hat{\mathbf{m}} + \mathbf{b}_m + \mathbf{n}_m)$$

where

- $\mathbf{b}_m$ is hard iron distortions (constant shifts in the sensor readings due to nearby permanent magnets or ferromagnetic materials)
- S_m is soft iron distortions, which are transformations (scaling and rotation effects) caused by surrounding metallic structures. It is a soft-iron distortion matrix, representing anisotropic distortions caused by the environment

The white Gaussian noise in the magnetometer is modelled as:

$$\mathbf{n}_m \sim \mathcal{N}(0, R_m)$$

where

- R_m is the noise covariance matrix defining the noise covariance along each axis and any correlation between axes

This noise has zero mean and constant variance over time.

Mathematical model for barometer

The mathematical model for a barometer is:

$$\mathbf{b} = \hat{\mathbf{b}} + \mathbf{b}_b \ + \mathbf{n}_b$$

where:

- $\mathbf{b}$ is the measured altitude from the barometer
- $\hat{\mathbf{b}}$ is the latent ideal altitude we wish to recover
- $\mathbf{b}_b$ is the barometer bias which changes over time and other factors such as temperature
- $\mathbf{n}_b$ is the white gaussian barometer noise, i.e. random-walk zero-mean measurement noise with variance σ_b^2

Mathematical model for odometry

Odometry refers to the use of data from actuators (wheels, treads, etc.) to estimate the overall motion of the vehicle.

The basic concept is to develop a model of how commanded motions of the vehicles wheels/joints/etc. induce motion of the vehicle itself and then to integrate these commanded motions over time in order to develop a model of the pose of the vehicle as function of time.

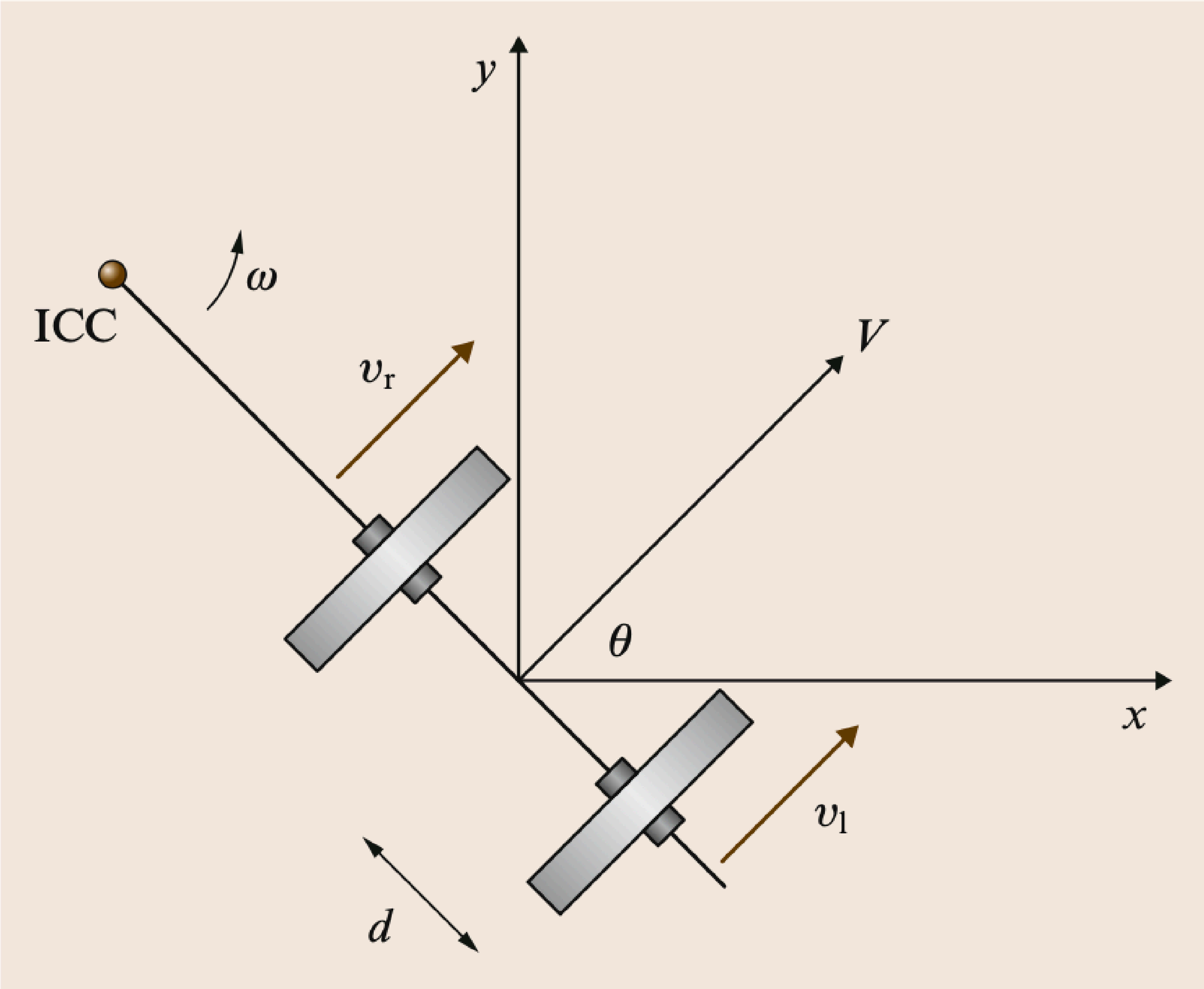
This is known as dead reckoning or deductive reckoning.

The details of odometry vary with vehicle design.

Odometry for differential drive

The simples is differential drive. i.e. a vehicle with two driveable wheels which are independently controlled and which are mounted along a common axis.

Assuming fixed location of the wheels on the vehicle, then for the wheels to remain in constant contact with the ground, the wheels must describe arcs on the plane such that the vehicle rotates around a point that lies on the wheels' common axis, the ICC i.e. instantaneous centre of curvature.



If the ground contact speeds of the left and right wheels are v_L and v_R respectively, the wheels are separated by a distance $2d$, and R describes the distance from the centre of the robot to the ICC, then:

$$\omega(R+d) = v_R$$
$$\omega(R-d) = v_L$$

Then:

$$\omega = \frac{v_R + v_L}{2d}$$
$$R = d\frac{v_R + v_L}{v_R - v_L}$$

The instantaneous velocity of the point midway between the robot's wheels is given by:

$$V = \omega R$$

Thus:

$$\theta(t) = \int \omega(t)\, dt$$
$$x(t) = \int V(t) \cos\left[\theta(t)\right]\, dt$$
$$x(t) = \int V(t) \sin\left[\theta(t)\right]\, dt$$

Given such a model and complete knowledge of the control inputs, we should, in principle, be able to estimate a robot’s pose at any time.

However errors in modelling the vehicle, uncertainty about control inputs, errors between commanded wheel rotation and true rotation, introduces error between the dead reckoning estimate of the vehicle motion and its true motion.

The problem of correcting for this error is called pose maintenance and requires integration of the dead reckoning estimate with estimates obtained from other sensor systems.

Odometry for quadcopters

We can not use rotor speeds or motor encoders of quadcopters for odometry similar to how we use wheel encoders for ground robots.

This is because wheel odometry relies in a direct predictable interaction between the wheels and ground. Quadcopters move in a fluid medium where forces like drag, wind and turbulence make it difficult to map motor speeds to precise movement.

Rotor speeds to not directly translate to displacement like wheel rotations do. This is because of aerodynamics, external forces and motor mixing equations.

Satellite-based positioning (GPS and GLS)

Global navigation satellite system (GNSS) and its most common instance global positional system (GPS) is the most commonly used mechanism for location estimation.

Anywhere on the Earth's surface GPS provides a three dimensional position estimate in absolute coordinates as well as current time and date.

GPS is based on received radio signals transmitted by an ensemble of satellites orbiting the earth. By comparing the time delays from the different satellite signals, a position fix can be computed.

The two most widely accepted GPS systems are NAVSTAR (operated by the US Air Force) and GLONASS (operated by the Russian government). Technically, though, GPS always refers to NAVSTAR.

The GPS network is based on a base constellation of 24 orbiting satellites along with up to six suplementary additional satellites that are also operational.

Their orbits are selected so that from almost any point of the earth's surface there will always be four or more satellites directly visible - a criterion for obtaining a GPS position estimate.

Each satellite repeatedly broadcasts a data packet known as coarse-acquisition (C/A) code which is received by the GPS receiver on the L1 channel at 1575.42MHz. The simple principle is that if the receiver knows the absolute positions of the observed satellites, the receiver position can be directly determined.

If the signal propagation time for the radio signals were known, the receiver position could be computed directly via trilateration. But this mean we would need highly accurate atomic clocks on all devices so instead only the satellites have these. The receiver computes the difference in signal propagation times between the different satellites, and uses this to compute a range estimate referred to as a pseudo-range (i.e. it has several sources of measurement noise). This geometric problem is referred to as multilateration or hyperbolic positioning and the solution is computed using a Kalman filter within the GPS receiver.

To avoid retaining an ephemeris (pose) table for the satellite broadcasts its own position and an accurate time signal as part of the data packet that it transmits.

Satellites broadcast at several different frequencies known as L1 through L5.

The standard service offered by NAVSTAR is determined by the L1 signal, which contains two unencrypted components:
- C / A (course acquisition)
- Navigation data message

It is also possible to use the encrypted L2 signal as well, even without the secret decryption keys, to provide augmented error correction (by observing the relative effects of ionospheric distortion as a function of frequency)

The restricted access signal broadcast on the L1 and L2 is known as the P-code, which is known as the Y-code or P(Y) once it is encrypted.

Both the C/A and P(Y) code include the navigation message stream that specifies clock bias data, orbital information, ionospherical propogation corrections factors, ephemeris data, status information on all the satellites, universal time code and other information.

GPS signals are in the microwave band and, as such, they can pass through plastic and glass, but are absorbed by water (wood, heavy foliage) and are reflected by many materials. As a consequence, GPS is unreliable in heavy forest, deep canyons, inside automobiles and boats, heavy snowfall or between tall buildings.

Mathematical Model

The measurement model for GPS is:

$$\mathbf{g} = \hat{\mathbf{g}} + \mathbf{n}_{GPS}$$

- $\mathbf{g}$ is the measured position from the GPS
- $\hat{\mathbf{g}}$ is the latent ideal position we wish to recover
- $\mathbf{n}_{GPS}$ is the white gaussian GPS noise, i.e. random-walk zero-mean measurement noise with variance σ^2_{GPS}

Wide Area Augmentation System (WAAS)

WAAS is an augmentation system operated by the FAA.

It is based around a supplementary signal that can be received by GPS receivers to improve their accuracy from 10-12m with GPS alone to between 1-2m.

The WAAS signal contains corrections for the GPS signal that reduce the effects of errors due to timing errors, satellite position corrections, and local perturbations due to variations in the ionosphere.

These correction terms are estimated by ground-based stations at fixed and accurately known positions and uplinked to satellites which broadcast them to suitably enabled GPS receivers.

WAAS is the US version but Europe, Japan and other Asian countries have it.

Differential GPS (DGPS)

DGPS is a technique for correcting GPS signals by using a nearby GPS receiver located at a known accurately-surveyed position. DGPS uses the same principles as WAAS but on a local scale without restoring to the use of satellite uplinks.

The receiver at the known position computes the error in the GPS signal and transmits it to the nearby receiver at the unknown location.

GPS Serial Protocols

NMEA supports an ASCII mode of communications based on a talker (a GPS receiver) and one or more listeners (computers), which receive simple protocol strings called sentences.

GPS-IMU integration

It has become common for GPS and IMU to be integrated into a single GPS-aided inertial navigation system (GPS/INS) to leverage their complementary advantages.

GPS doesn't solve all the problems associated with robot pose estimation. It does not directly obtain information about vehicle orientation. Finally, it is not always possible to obtain a GPS fix.

Range sensing

Range sensors are devices that capture the three-dimensional structure of the world from the viewpoint of the sensor, usually measuring the depth to the nearest surfaces. These measurements could be at a single point, across a scanning plane, or a full image with depth measurements at every point.

The benefits of this range data is that a robot can be relatively certain where the real world is, relative to the sensor, thus allowing the robot to more reliably find navigable routes, avoid obstacles, act on industrial parts, etc.

LiDAR, radar, ultrasonic sensors and cameras have their own niche sets of benefits. Usually multiple of these technologies are used to create a long and short range map of a vehicle's surroundings under a range of weather and lighting conditions.

With sensor data fusion it is important that these technologies complement each other but also that they have sufficient overlap in order to increase redundancy and improve safety.

Ultrasonic waves suffer from strong attenuation in air beyond a few meters, therefore ultrasonic sensors are primarily used for short-range object detection.

Cameras are cost-efficient and easily available but require significant processing to extract useful information and depend strongly on ambient light conditions. However, cameras are unique in that they are only technology that can "see colour".

LiDAR and radar can map surroundings as well as measure object velocity.

Range images and point sets

Range data is a two-and-half dimension or three-dimensional representation of the scene around the robot.

The three-dimensionality arises because we are measuring the (X, Y, Z) coordinates of one or more points in the scene. Often only a single range image is used at each time instance. This means that we only observe the front sides of objects. So we don't have a full 3-D observation, but a 2.5D observation.

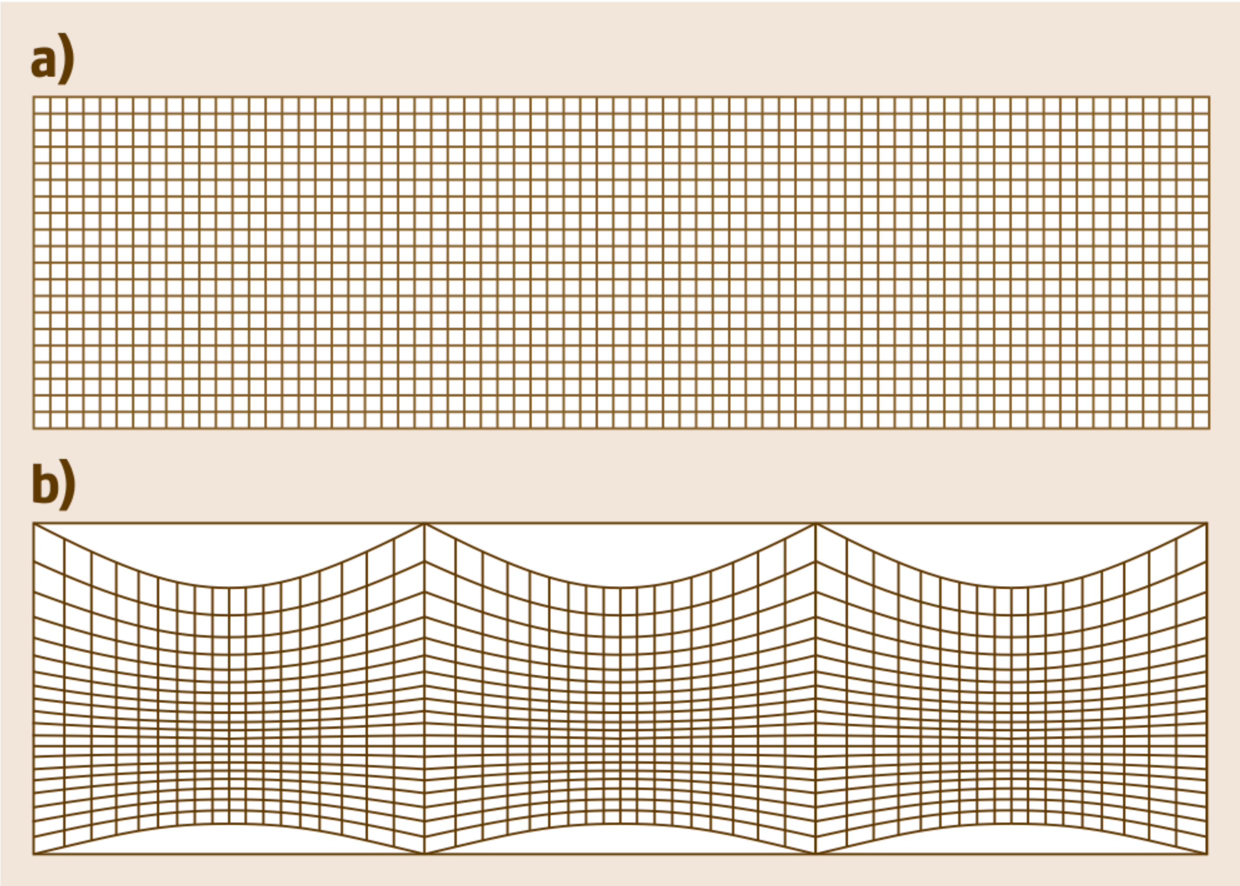
There are two standard formats for representing range data. The first is an image

$$d(i, j)$$

which records the distance d to the corresponding scene point (x, y, z) for each image pixel (i, j). There are several common mappings:

$$f(i, j, d(i, j)) \rightarrow (X, Y, Z)$$

The most common image mappings are illustrated below:

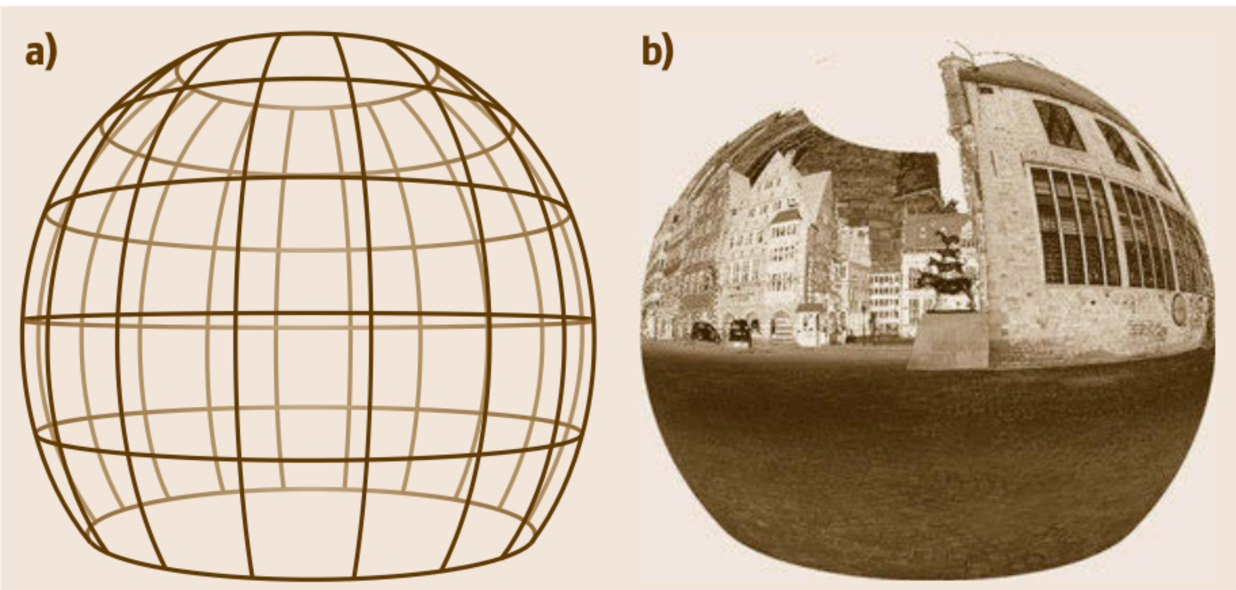


a) shows a equirectangular projection of a spherical range scan

b) shows a rectilinear projection with three images combined to unwrap a scanning sphere

Range scanners with rotating mirrors sample in spherical coordinates, i.e. (θ, ϕ, d).

a) shows a sphere as a longitude and latitude mesh and b) it superimposed with the reflectance values of the image above.

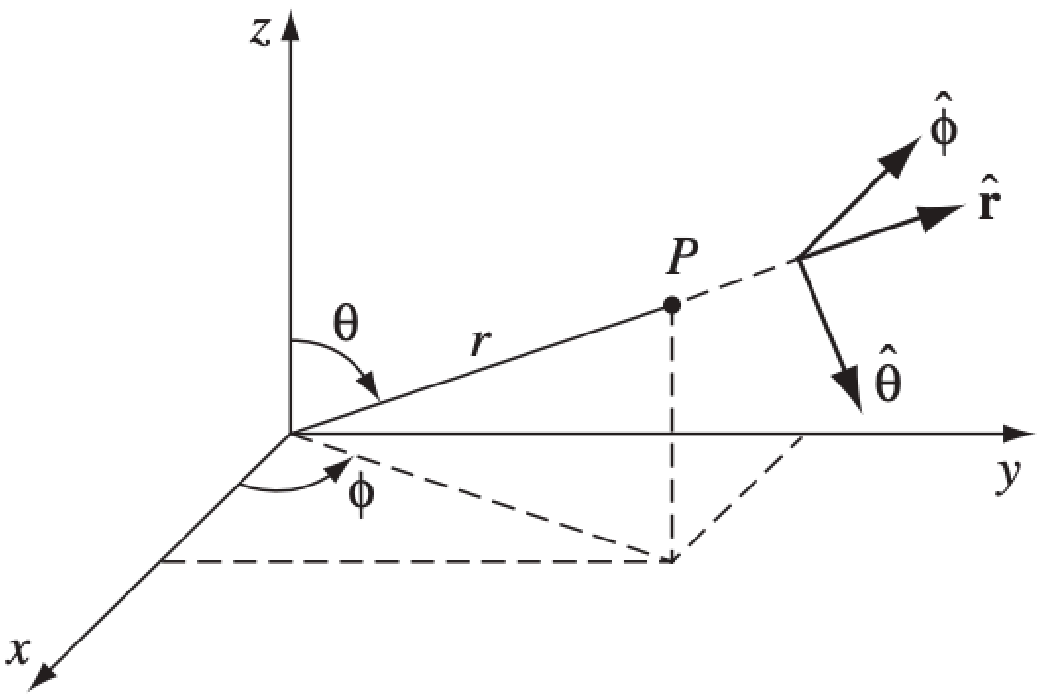


Range sensors typically rotate or scan in a way that naturally aligns with spherical coordinates. Rotating LiDAR spin around a central axis, naturally measuring angles θ (horizontal/azimuth) and ϕ (vertical/elevation).

If you imagine bending out the sphere b) above into the rectilinear projection with three images combined to unwrap a scanning sphere then you can see that the most natural range image representation is to put on the axis of the range image, i.e.

$$i = \theta$$
$$j = \phi$$

Imagine shining $d(i, j)$ as the equivalent of what r stands for in spherical coordinates into the image. Then it makes sense for the i axis of pixels to represent the spanning of the angle θ and for the j axis of pixels to represents the spanning of the angle ϕ.



This is a projection of the samples sphere to a 2-D image, which includes a distortion. The mapping

$$f(i, j, d(i, j)) \rightarrow (X, Y, Z)$$

is thus:

$$\begin{aligned} d &= \sqrt{x^2 + y^2 + z^2} \\ \theta &= \arccos\left(\frac{z}{r(\text{dont know why this isnt d})}\right) \quad (cah) \\ \phi &= \arctan\left(\frac{y}{x}\right) \quad (toa) \end{aligned}$$

Cartesian coordinates may be retrieved from the spherical coordinates by

$$
\begin{aligned}
\tan(\phi) &= \frac{y}{x} \\
\frac{\sin(\phi)}{\cos(\phi)} &= \frac{y}{x} \\
d &= \sqrt{x^2 + x^2\,\frac{\sin^2(\phi)}{\cos^2(\phi)} + d^2\cos^2(\theta)} \\
d^2\cos^2(\phi) &= x^2\cos^2(\phi) + x^2\ \sin^2(\phi) + d^2\cos^2(\theta)\cos^2(\phi) \\
d^2\cos^2(\phi) &= x^2\cos^2(\phi) + x^2\ (1-\cos^2(\phi)) + d^2(1-\sin^2(\theta))\cos^2(\phi) \\
d^2\cos^2(\phi) &= x^2\cos^2(\phi) + x^2 - x^2\cos^2(\phi) + (d^2 - d^2\sin^2(\theta))\cos^2(\phi) \\
d^2\cos^2(\phi) &= x^2\cos^2(\phi) + x^2 - x^2\cos^2(\phi) + d^2\cos^2(\phi) - d^2\sin^2(\theta)\cos^2(\phi) \\
0 &= 0 + x^2 + 0 + 0 - d^2\sin^2(\theta)\cos^2(\phi) \\
x &= d\sin(\theta)\cos(\phi)
\end{aligned}
$$

$$
\begin{aligned}
\frac{\sin(\phi)}{\cos(\phi)} &= \frac{y}{x} \\
y &= d\sin(\theta)\cos(\phi)\frac{\sin(\phi)}{\cos(\phi)} \\
y &= d\ \sin(\theta)\ \sin(\phi)
\end{aligned}
$$

$$
\begin{aligned}
\cos(\theta) &= \frac{z}{d} \\
z &= d\cos(\theta)
\end{aligned}
$$

where:

- d is the depth measurement from the sensor
- θ is the azimuth angle
- ϕ is the elevation angle

The other common projection is the perspective one, usually called rectilinear projection. This includes range images produced by kinetic-like devices.

In the perspective projection we normalize by distance so that we remove explicit dependence on d.

Here, the range values are assumed to be projected in a pinhole camera like fashion. The pinhole camera model assumes that all rays pass through a single point (optical centre) and are projected onto a flat image plane.

If the camera has a focal length f:

$$
\begin{aligned}
i &= f\frac{X}{Z} \\
j &= f\frac{Y}{Z}
\end{aligned}
$$

For convenience in computer vision, this focal length is often set to 1 or absorbed into a scaling factor later:

$$
\begin{aligned}
i &= \frac{X}{Z} \\
j &= \frac{Y}{Z}
\end{aligned}
$$

Thus, expression this in spherical coordinates, we get:

$$
\begin{aligned}
X &= d\sin(\theta)\cos(\phi) \\
Y &= d\sin(\theta)\sin(\phi) \\
Z &= d\cos(\theta)
\end{aligned}
$$

$$
\begin{aligned}
i &= \frac{d\sin(\theta)\cos(\phi)}{d\cos(\theta)} = \frac{\sin(\theta)\cos(\phi)}{\cos(\theta)} \\
j &= \frac{d\sin(\theta)\sin(\phi)}{d\cos(\theta)} = \frac{\sin(\theta)\sin(\phi)}{\cos(\theta)}
\end{aligned}
$$

these are basic equations for rectilinear projection

Real sensors don't align with the standard spherical coordinate system. We can't assume that its forward direction is $\theta = \phi = 0$, instead

- cameras have a central azimuth, i.e. a longitude offset θ_0
- cameras have a central elevation i.e. a latitude offset ϕ_1
  so we say that our sensor is centerd at:

  $$(\theta_0, \phi_1)$$

  and we need to correct for this.

The new x-axis is shifted by the angle θ_0 meaning we replace θ with $\theta - \theta_0$:

$$\sin\theta \to \sin(\theta - \theta_0)$$

The new z-axis is affected by ϕ_1 which tilts the frame

If the sensor is perfectly aligned, the depth coordinate is just:

$$Z = d\cos(\theta)$$

Since the sensor is tilted by ϕ_1 instead of looking forward, the sensor looks in a direction already rotated by ϕ_1

There is a vertical contribution and azimuth contribution to this tilt:

$$Z' = Z'_{\text{vertical}} + Z'_{\text{azimuth}}$$

Using the trigonometric projection, the contribution to Z' due to pure vertical alignment is:

$$Z'_{\text{vertical}} = d\sin(\phi_1)\sin(\phi)$$

if:

- $\phi_1 = 0$, (no tilt) this term disappears

- $\phi_1 = \frac{\pi}{2}$, the sensor is fully vertical and this term dominates

Using the trigonometric projection, the horizontal contribution to Z' due to pure horizontal alignment is:

$$\begin{aligned}
Z_{\text{horizontal}} &= d_{\text{horizontal}} \cos(\theta) \\
Z'_{\text{horizontal}} &= d_{\text{horizontal}} \cos(\theta - \theta_0) \\
Z'_{\text{horizontal}} &= d \cos(\phi) \cos(\theta - \theta_0) \\
Z'_{\text{azimuth}} &= d \cos(\phi_1) \cos(\phi) \cos(\theta - \theta_0)
\end{aligned}$$

This means that:

$$\begin{aligned}
Z' &= Z'_{\text{vertical}} + Z'_{\text{azimuth}} \\
Z' &= d(\sin(\phi_1)\sin(\phi) + \cos(\phi_1)\cos(\phi)\cos(\theta - \theta_0))
\end{aligned}$$

The projection proceeds as:

$$\begin{aligned}
i &= \frac{X'}{Z'} \\
&= \frac{d\cos(\phi)\sin(\theta - \theta_0)}{d(\sin(\phi_1)\sin(\phi) + \cos(\phi_1)\cos(\phi)\cos(\theta - \theta_0))} \\
&= \frac{\cos(\phi)\sin(\theta - \theta_0)}{\sin(\phi_1)\sin(\phi) + \cos(\phi_1)\cos(\phi)\cos(\theta - \theta_0)}
\end{aligned}$$

$$\begin{aligned}
j &= \frac{Y'}{Z'} \\
&= \\
&= \frac{\cos(\phi_1)\sin(\phi) - \sin(\phi_1)\cos(\phi)\cos(\theta - \theta_0)}{\sin(\phi_1)\sin(\phi) + \cos(\phi_1)\cos(\phi)\cos(\theta - \theta_0)}
\end{aligned}$$

where:

- θ_0 is the central longitude
- ϕ_1 is the central latitude

Some range sensors only record distances in a slice, so the scene (x, y) is represented by the linear image $d(i)$ for each pixel i

The second format is as a list $\{(x_i, y_i, z_i)\}$ of 3D data points but this format can be used with all of the mappings listed above. Given the conversions from the image data $d(i, j)$ to (x, y, z) the range data is only supplied as a list, which is usually called the point cloud.

Triangulation

Triangulation sensors measure depth by determining the angle formed by the rays from a world point to two sensors. The sensors are separated by a baseline of length b, which forms the third segment of a triangle between the sensors and the point.

For simplicity, let one of the rays form a right angle with the baseline. Then the angle θ of the other sensor ray is related to the depth Z perpendicular to the baseline by

$$\tan(\theta) = \frac{Z}{b}$$

An image sensor measures the angle θ by an offset on an image plane from the primary ray, this offset x is called the disparity.

If we assume the image plane is parallel to the baseline, then $\tan(\theta) = \frac{f}{x}$ and we get the basic equation for triangulation depth sensors:

$$Z = \frac{fb}{x}$$

An important concept is the depth resolution of a sensor, how precisely can the sensor measure the depth?

Differentiating with respect to x and substituting for x gives:

$$\frac{dZ}{dx} = \frac{-Z^2}{fb}$$

Triangulation precision falls off the square of the distance to an object.

Time of Flight sensors

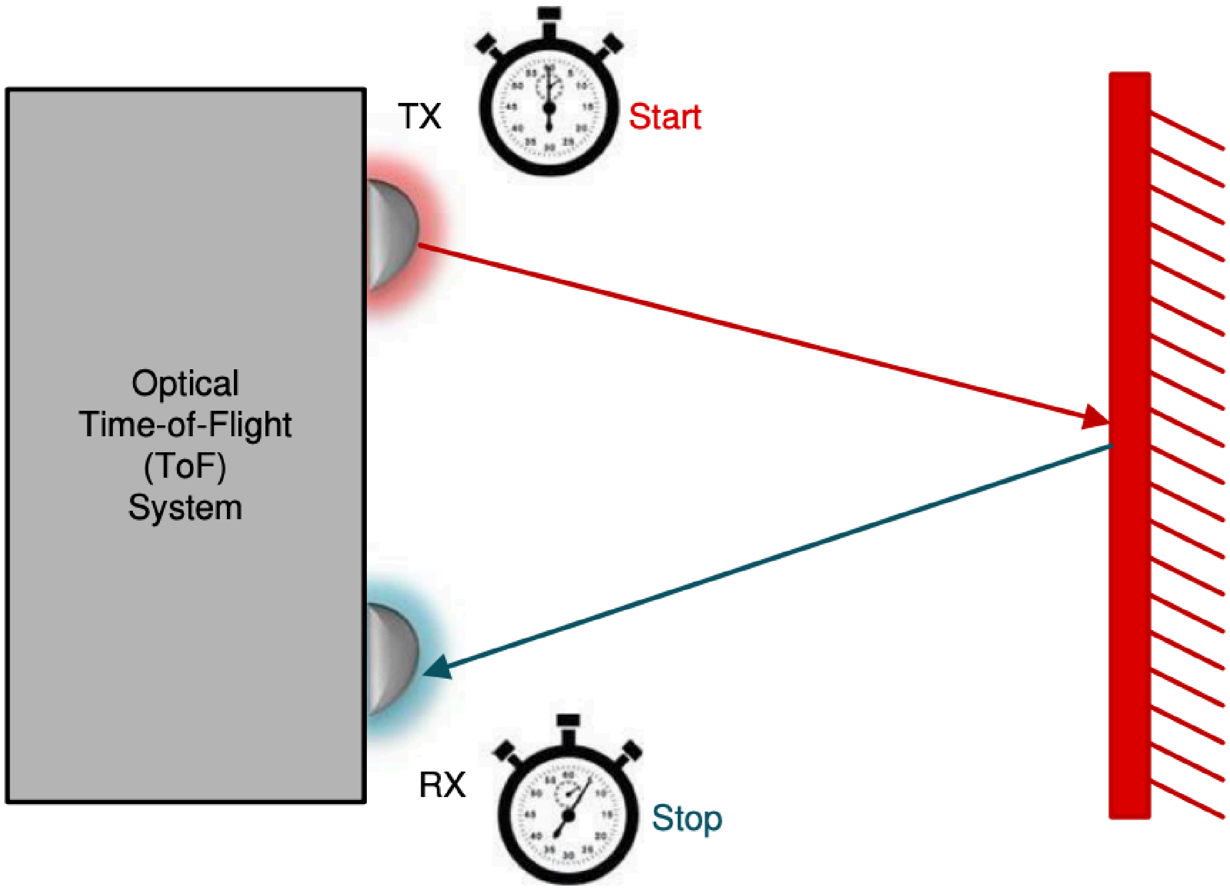
The principle behind time-of-flight (TOF) sensors is like that of radar: measure the time it takes for light to be projected out to an object and return.

Because they measure time of flight, these sensors can theoretically have constant precision in depth measurement, no matter how far an object, but cannot duplicate the very fine precision of triangulation sensors for close objects, and hence are not used in close-range metrology applications.

Direct TOF (LiDAR)

In direct TOF sensors, travel time is measured by a high-speed chronometer. Laser-based direct TOF range sensors are also called LiDAR (light detection and ranging) or LADAR (laser radar).

Optical time-of-flight LiDAR consist of a light transmitter, usually in the form of a laser, and a light receiver. These systems measure distance by emitting a pulse of light onto an object and receiving the reflected pulse of light from the object. The time it takes for the light to travel to and from the object can be used to calculate the distance between the transmitter, receiver and the object.



Travel time is multiplied by the speed of light in the given medium (space, air or water and adjusted for the density and temperature of the medium):

$$2d = ct$$

where

- d is the distance to the object
- c is the speed of light
- t is the measured travel time

Error in measuring the time t gives rise to a proportional distance error. In practice, the peak of the output pulse, which has a finite extent, is measured. Weak reflections from distant objects make it harder to measure this peak and thus the error increases with distance. Averaging multiple readings can reduce the random error in these readings.

The simplest TOF sensors transmit only a single beam, thus range measurements are only obtained from a single surface point. But we usually need from info, so the range data is usually supplied as a vector of ranges to surface lying in a plane or as an image.

To obtain these denser representations, the laser beam is swept across the scene. Scanning or averaging requires multiple TOF pulses at high repetition rates, which can give rise to ambiguity about which pulse is actually being received.

If δt is the time between pulses, then the ambiguity interval for the device is

$$\frac{1}{2c\delta t}$$

e.g. for a pulse interval repetition rate of $100kHz$, the ambiguity interval is $1500m$. So $1500m$ is the maximum unambiguous range that a LiDAR system can measure before ambiguity arises due to pulse repetition

Multiple-beam scanning LiDARs can increase the amount of information available.

Stereo vision