Principles of Robot Autonomy I

Robotic sensors and introduction to computer vision

Stanford University

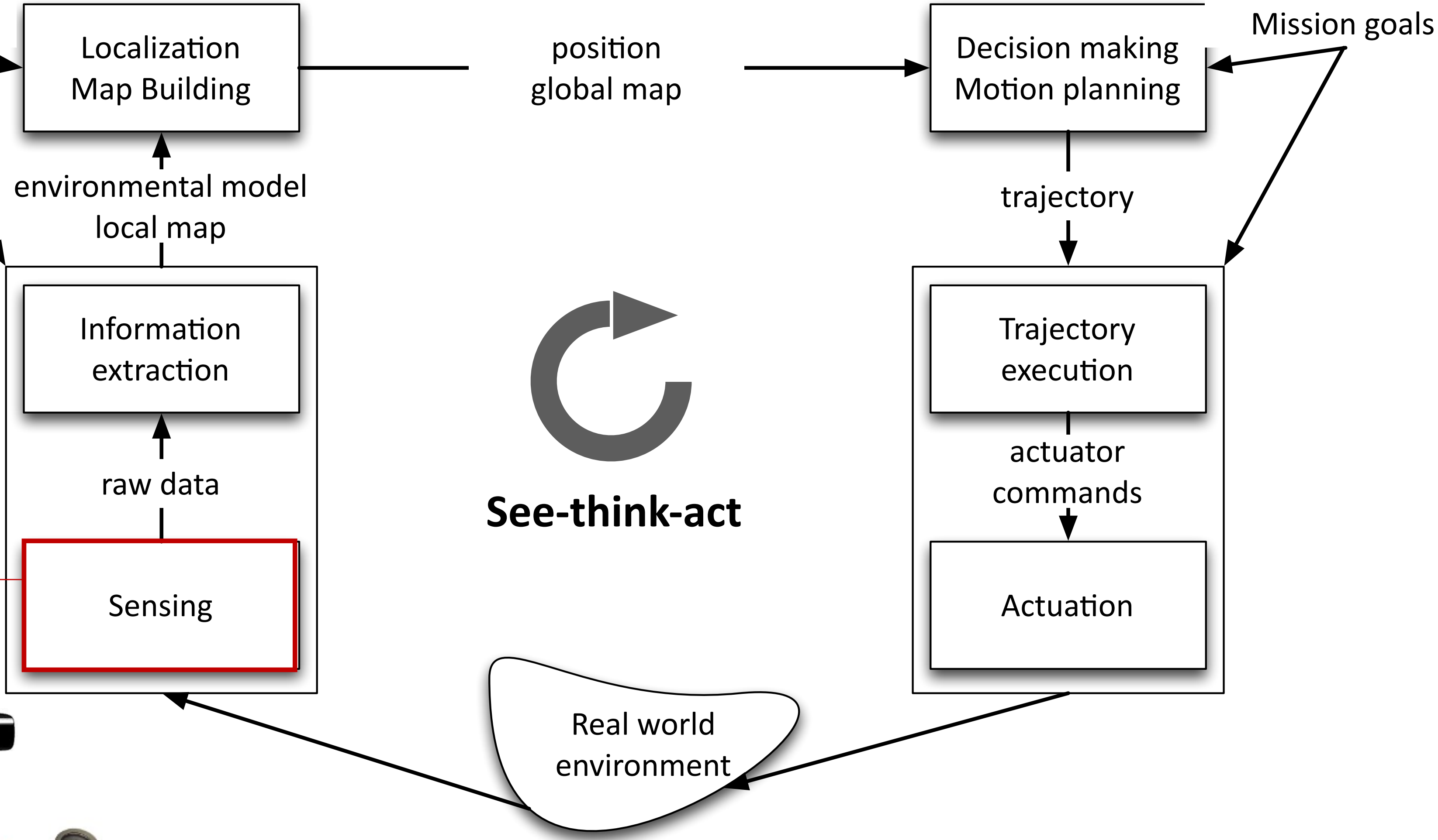
Autonomous Systems Lab

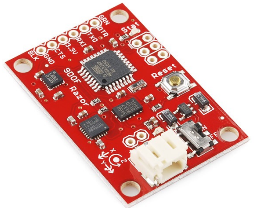
Agenda

- Agenda
  - Overview of key performance characteristics for robotic sensors
  - Overview of main sensors for robot autonomy, e.g. proprioceptive / exteroceptive, passive / active
  - Intro to computer vision

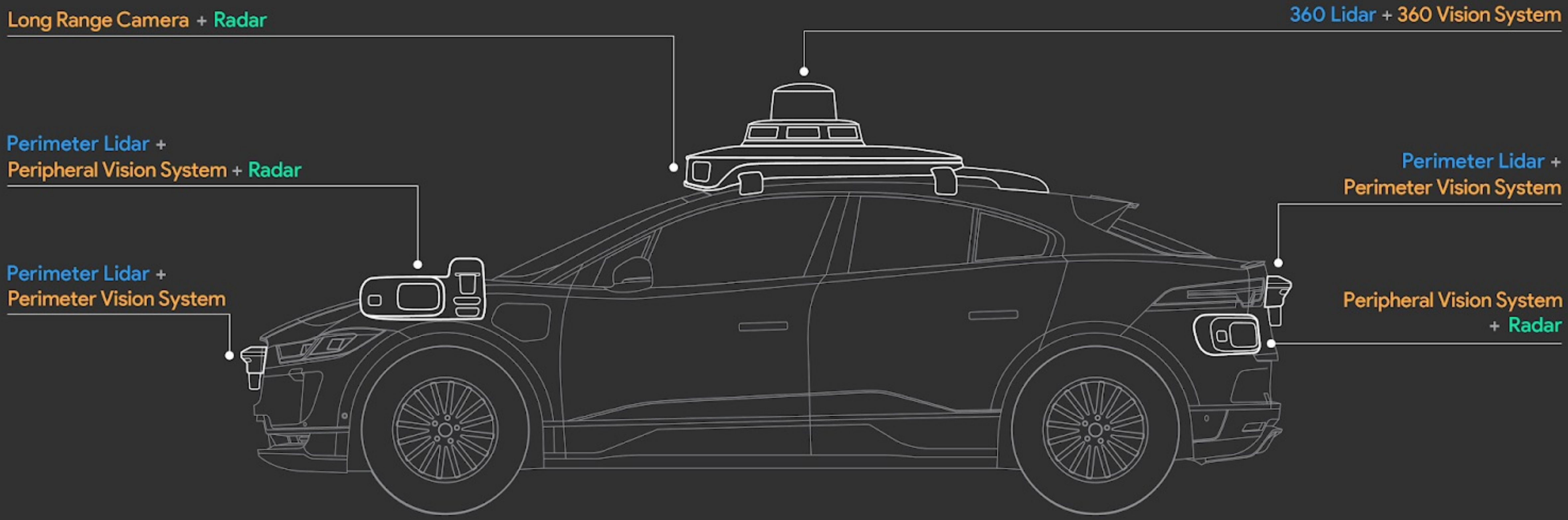
- Readings:
  - Chapters 7 and 8.1 in PoRA lecture notes

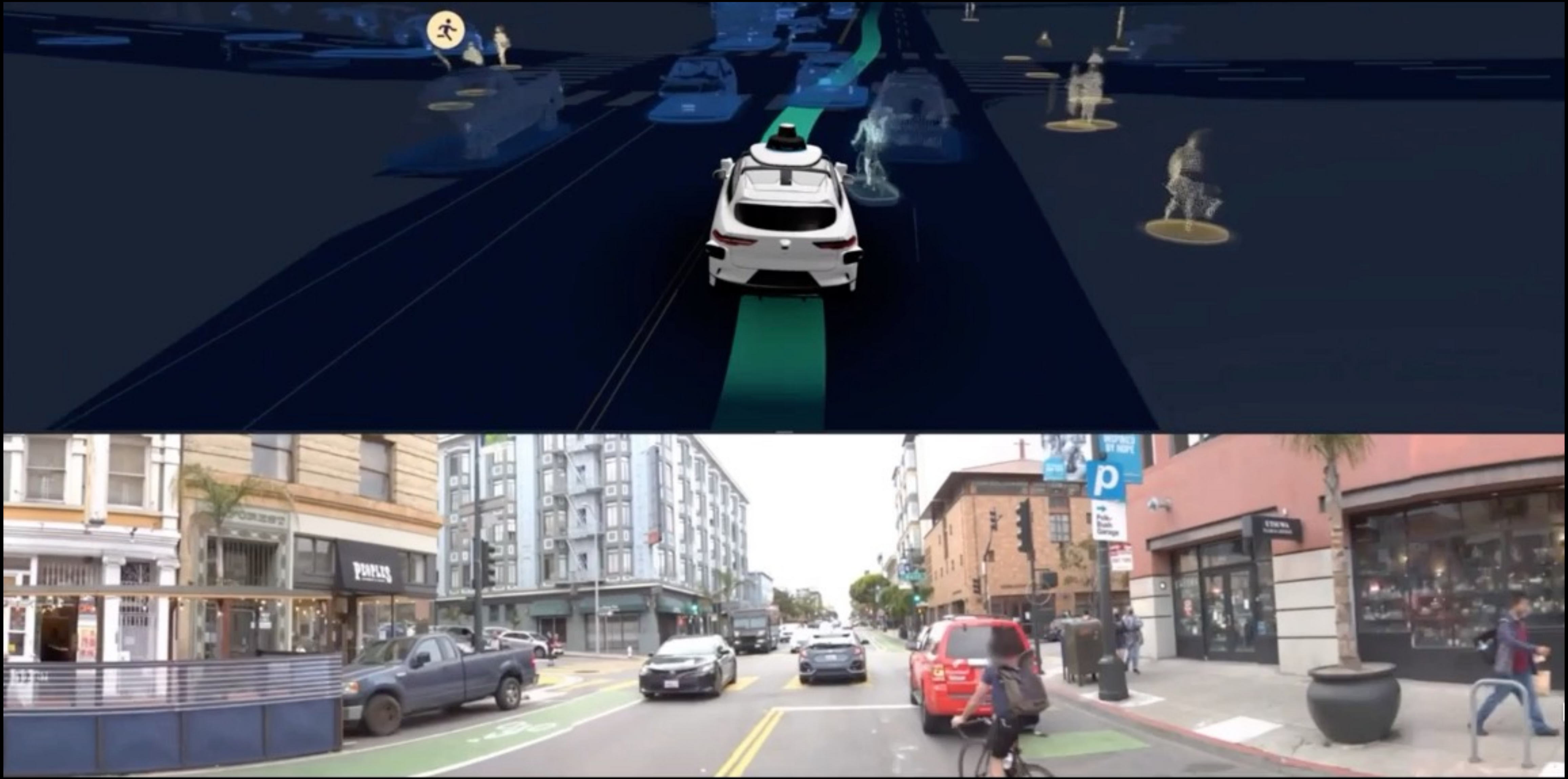
Sensors for mobile robots





Example: self-driving cars




Polk
Bush
Garage

Classification of sensors

- Proprioceptive: measure values internal to the robot
  - E.g.: motor speed, robot arm joint angles, and battery voltage
- Exteroceptive: acquire information from the robot's environment
  - E.g.: distance measurements and light intensity

- Passive: measure ambient environmental energy entering the sensor
  - Challenge: performance heavily depends on the environment
  - E.g.: temperature probes and cameras
- Active: emit energy into the environment and measure the reaction
  - Challenge: might affect the environment
  - E.g.: ultrasonic sensors and laser rangefinders

Sensor performance: design specs

- Dynamic range: ratio between the maximum and minimum input values (for normal sensor operation)
- Resolution: minimum difference between two values that can be detected by a sensor
- Linearity: whether or not the sensor’s output response depends linearly on the input
- Bandwidth or frequency: speed at which a sensor provides readings (in Hertz)

Sensor performance: in situ specs

- **Sensitivity**: ratio of output change to input change
- **Cross-sensitivity**: sensitivity to quantities that are unrelated to the target quantity
- **Error**: difference between the sensor output m and the true value v

$$\text{error} \coloneqq m - v$$

- **Accuracy**: degree of conformity between the sensor's measurement and the true value

$$\text{accuracy} \coloneqq 1 - |\text{error}|/v$$

- **Precision**: reproducibility of the sensor results

Sensor errors

- Systematic errors: caused by factors that can in theory be modeled; they are deterministic
  - E.g.: calibration errors
- Random errors: cannot be predicted with sophisticated models; they are stochastic
  - E.g.: spurious range-finding errors
- Error analysis: performed via a probabilistic analysis
  - Common assumption: symmetric, unimodal (and often Gaussian) distributions; convenient, but often a coarse simplification
  - Error propagation characterized by the *error propagation law*

An ecosystem of sensors

- Encoders
- Heading sensors
- Accelerometers and IMU
- Beacons
- Active ranging
- Cameras

Encoders

- Encoder: an electro-mechanical device that converts motion into a sequence of digital pulses, which can be converted to relative or absolute position measurements
  - proprioceptive sensor
  - can be used for robot localization

- Fundamental principle of optical encoders: use a light shining onto a photodiode through slits in a metal or glass disc

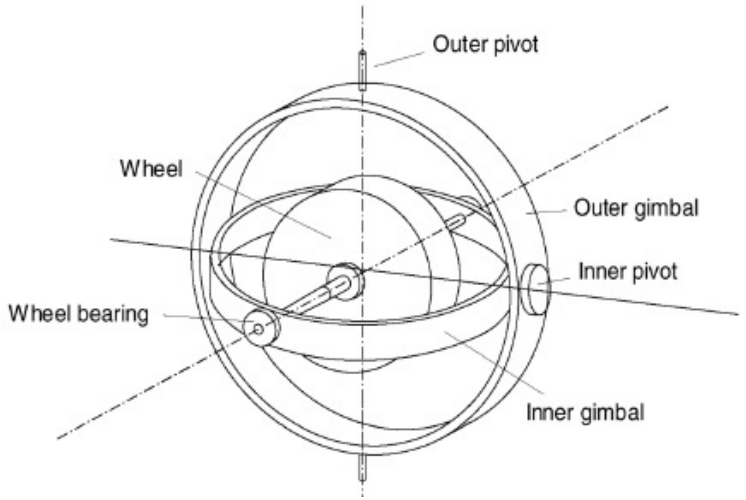
Wheel encoder
Credit: Pololu

Credit: Honest Sensor

Heading sensors

- Used to determine robot's orientation, it can be:
    1. Proprioceptive, e.g., gyroscope (heading sensor that preserves its orientation in relation to a fixed reference frame)
    2. Exteroceptive, e.g., compass (shows direction relative to the geographic cardinal directions)
- Fusing measurements with velocity information, one can obtain a position estimate (via integration) -> *dead reckoning*

- Fundamental principle of mechanical gyroscopes: angular momentum associated with spinning wheel keeps the axis of rotation inertially stable



Credit: SNS

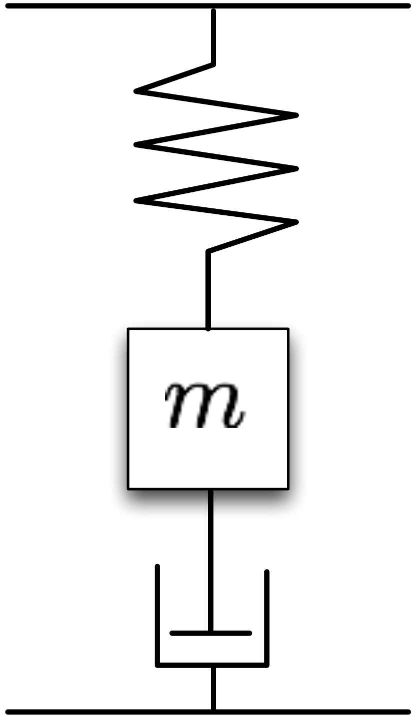
Accelerometer and IMU

- Accelerometer: device that measures all external forces acting upon it
- Mechanical accelerometer: essentially, a spring-mass-damper system

$$F_{\text{applied}} = m\ddot{x} + c\dot{x} + kx$$

with *m* mass of proof mass, *c* damping coefficient, *k* spring constant; in steady state

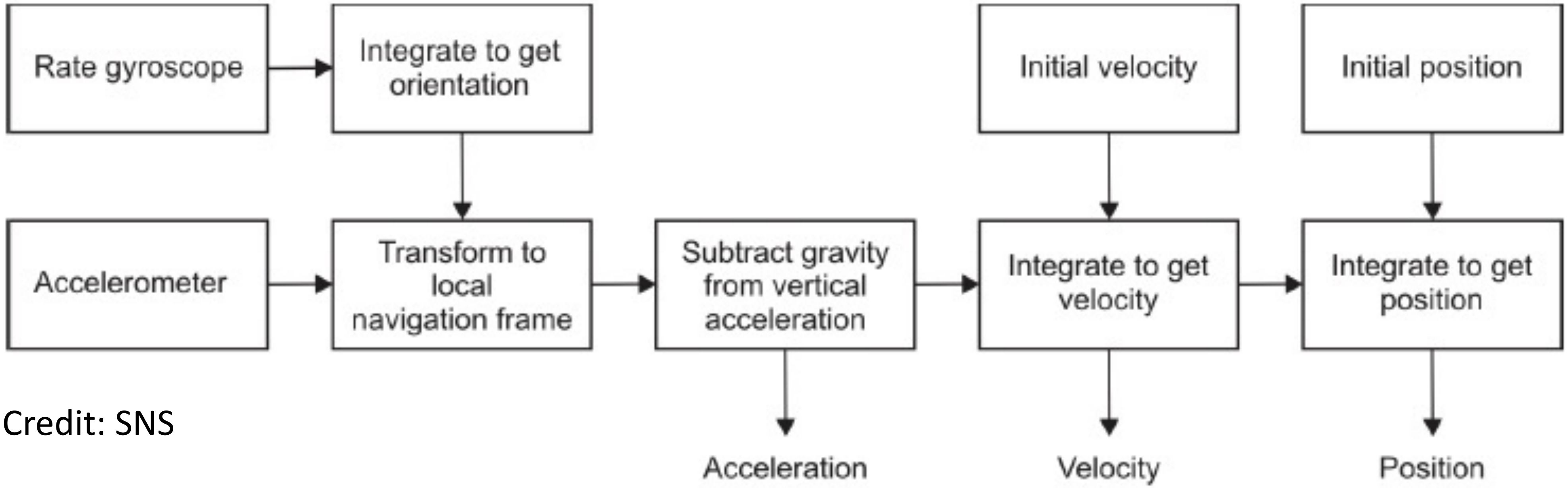
$$a_{\text{applied}} = \frac{kx}{m}$$



- Modern accelerometers use MEMS (cantilevered beam + proof mass); deflection measured via *capacitive* or *piezoelectric* effects

Inertial Measurement Unit (IMU)

- Definition: device that uses gyroscopes and accelerometers to estimate the relative position, orientation, velocity, and acceleration of a moving vehicle with respect to an inertial frame
- *Drift* is a fundamental problem: to cancel drift, periodic references to external measurements are required



Credit: SNS

Beacons

- Definition: signaling devices with precisely known positions
- Early examples: stars, lighthouses
- Modern examples: GPS, motion capture systems

Active ranging

- Provide direct measurements of distance to objects in vicinity
- Key elements for both localization and environment reconstruction
- Main types:
    1. Time-of-flight active ranging sensors (e.g., ultrasonic and laser rangefinder)

Credit: https://electrosome.com/hc-sr04-ultrasonic-sensor-pic/

2. Geometric active ranging sensors (optical triangulation and structured light)

Time-of-flight active ranging

- Fundamental principle: time-of-flight ranging makes use of the propagation of the speed of sound or of an electromagnetic wave
- Travel distance is given by

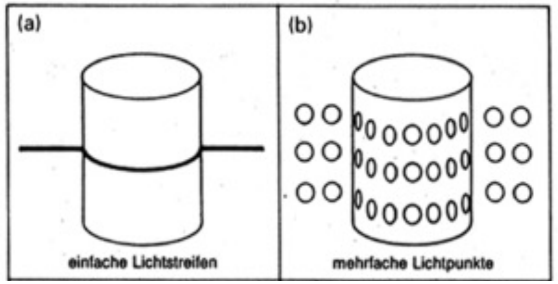
$$d = c\,t$$

where *d* is the distance traveled, *c* is the speed of the wave propagation, and *t* is the time of flight

- Propagation speeds:
  - Sound: 0.3 m/ms
  - Light: 0.3 m/ns
- Performance depends on several factors, e.g., uncertainties in determining the exact time of arrival and interaction with the target

Geometric active ranging

- Fundamental principle: use geometric properties in the measurements to establish distance readings
- The sensor projects a known light pattern (e.g., point, line, or texture); the reflection is captured by a receiver and, together with known geometric values, range is estimated via triangulation
- Examples:
  - Optical triangulation (1D sensor)
  - Structured light (2D and 3D sensor)



Credit: Matt Fisher

Several other sensors are available

- Classical, e.g.:
  - Radar (possibly using Doppler effect to produce velocity data)
  - Tactile sensors

- Emerging technologies:
  - Artificial skins
  - Neuromorphic cameras

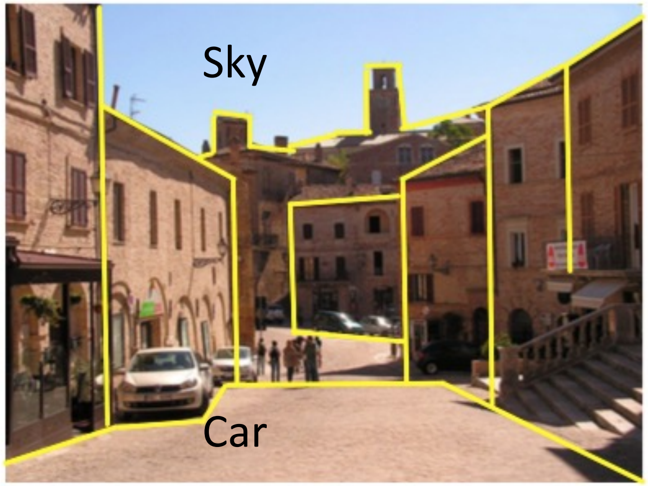
Introduction to computer vision

- Aim
  - Learn about cameras and camera models
  - Learn about the outputs of perception and what they might be used for

- Readings
  - Siegwart, Nourbakhsh, Scaramuzza. Introduction to Autonomous Mobile Robots. Section 4.2.3.
  - D. A. Forsyth and J. Ponce [FP]. Computer Vision: A Modern Approach (2nd Edition). Prentice Hall, 2011. Chapter 1.
  - R. Hartley and A. Zisserman [HZ]. Multiple View Geometry in Computer Vision. Academic Press, 2002. Chapter 6.1.

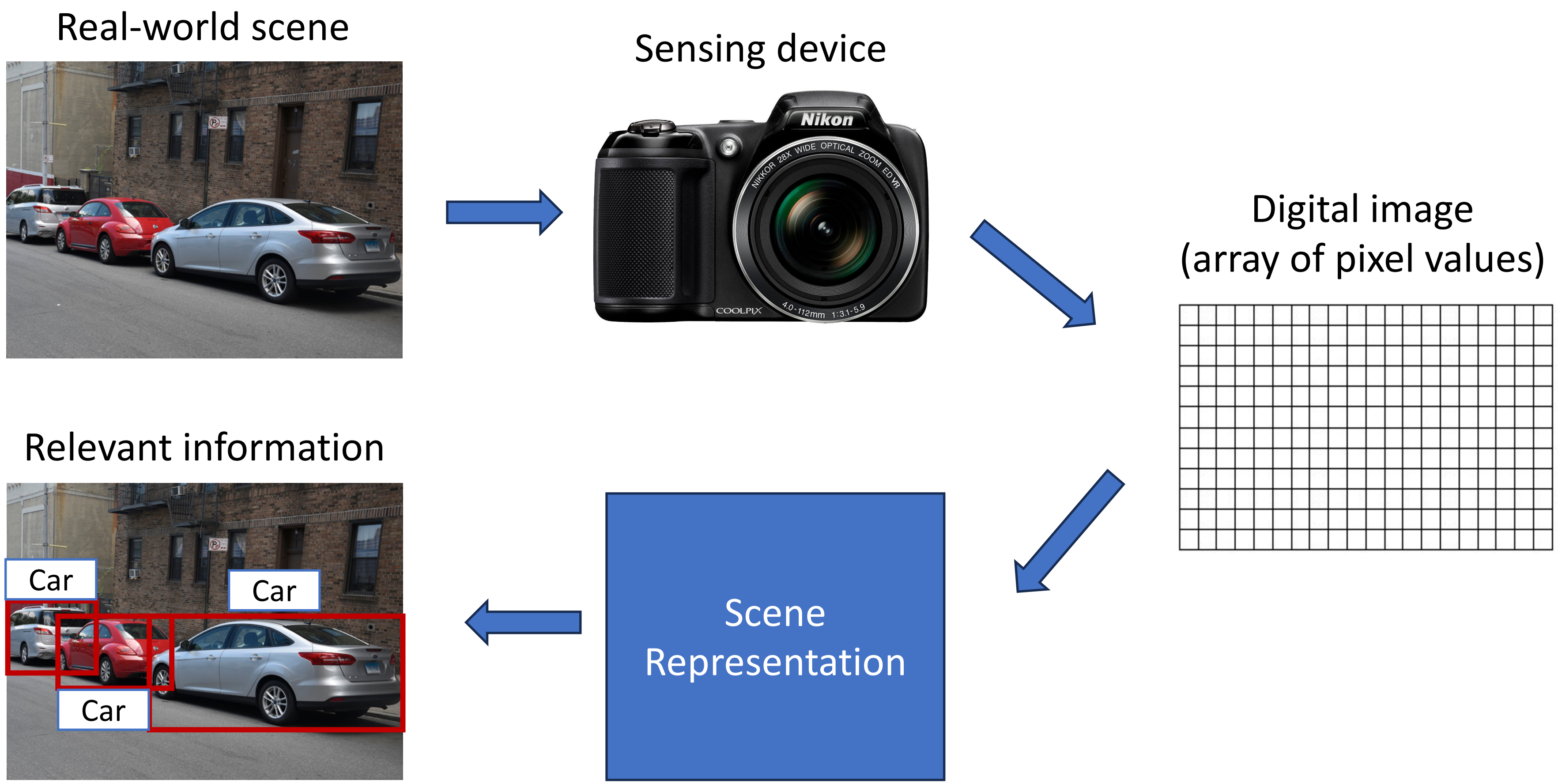
Vision

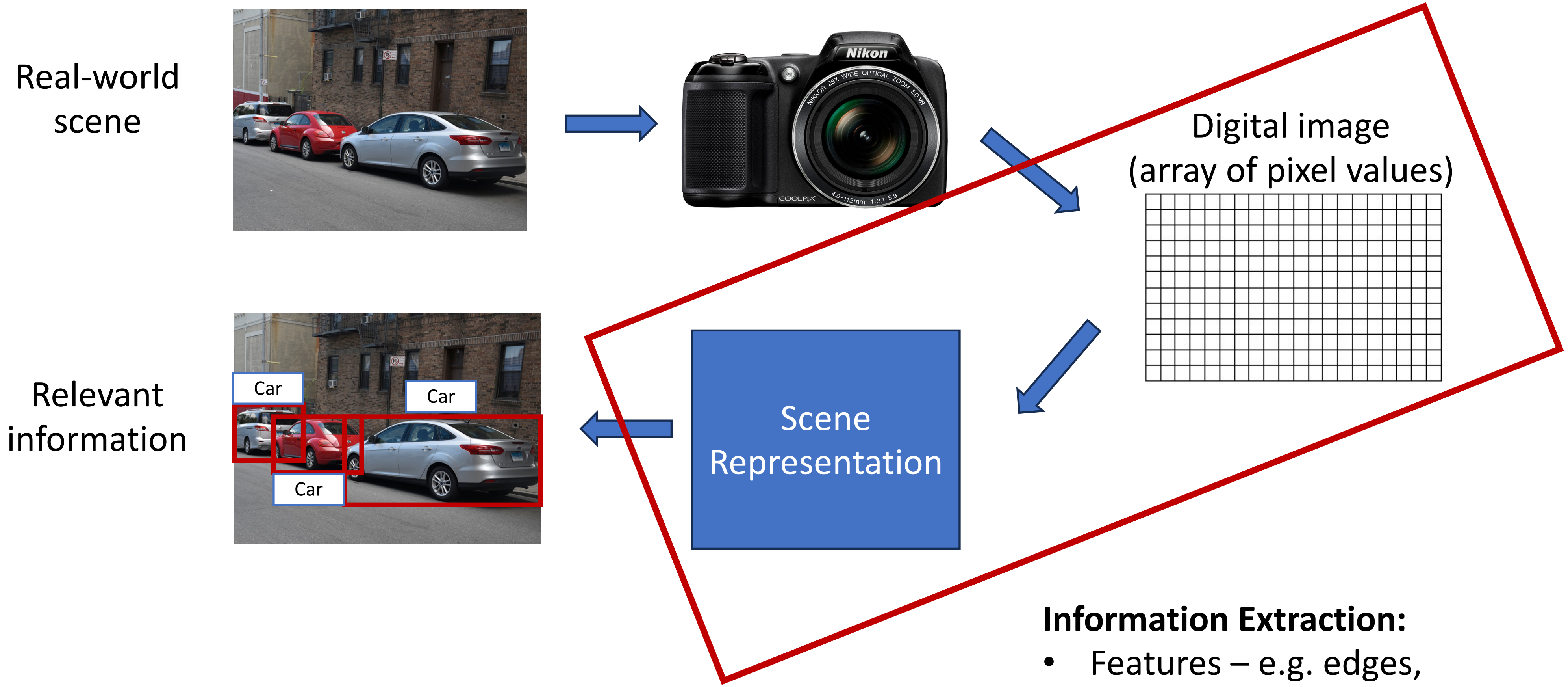
- Vision: ability to interpret the surrounding environment using light in the visible spectrum reflected by objects in the environment
- Human eye: provides enormous amount of information, ~millions of bits per second
- Cameras (e.g., CCD, CMOS): capture light -> convert to digital image -> process to get relevant information (from geometric to semantic)



1. Information extraction
2. Interpretation

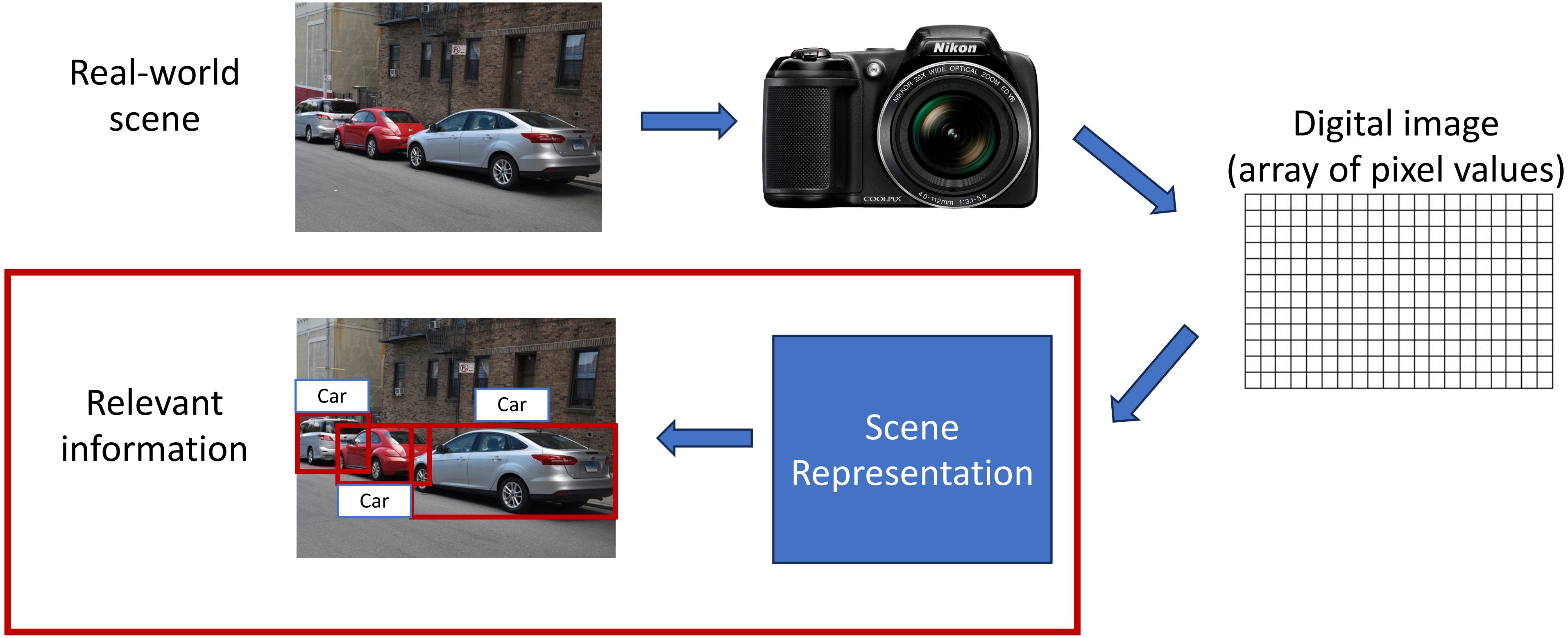
Computer Vision Pipeline





Information Extraction:

- Features – e.g. edges, corners, texture, colors, etc.
- 3D structure



Interpretation:
- Object detection
- Object tracking
- Image registration
- Image segmentation

Object Detection

- Goal: Detect instances of semantic objects of a certain class
  - E.g. pedestrian detection, face detection
- Approaches:
  - Traditional methods, e.g.:
    - Scale-invariant feature transform (SIFT)
    - Histogram of Oriented Gradients (HOG)
  - Learning-based:
    - Using region proposals
    - Without region proposals: You Only Look Once (YOLO), Single Shot Detector (SSD)

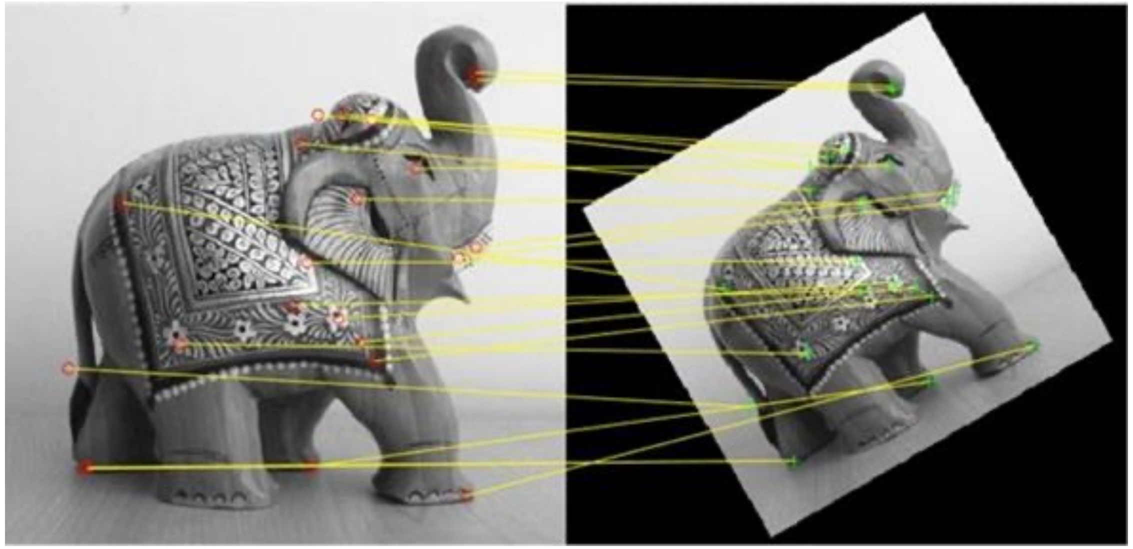
Object Tracking

- Goal: Follow and locate a specific object across a sequence of images or video frames
- Applications: Autonomous driving, surveillance, augmented reality, medical imaging, sports analysis, etc.
- Approaches:
  - Traditional methods, e.g. mean-shift tracking or Kalman filters
  - Learning-based methods, e.g. Siamese networks or recurrent neural networks (RNNs)

Image Registration

- Goal: Transform different sets of data into one coordinate system
- Examples:
  - Data from multiple photographs (e.g. with different viewpoints)
  - Data from different sensors (e.g. LIDAR and RGB camera)

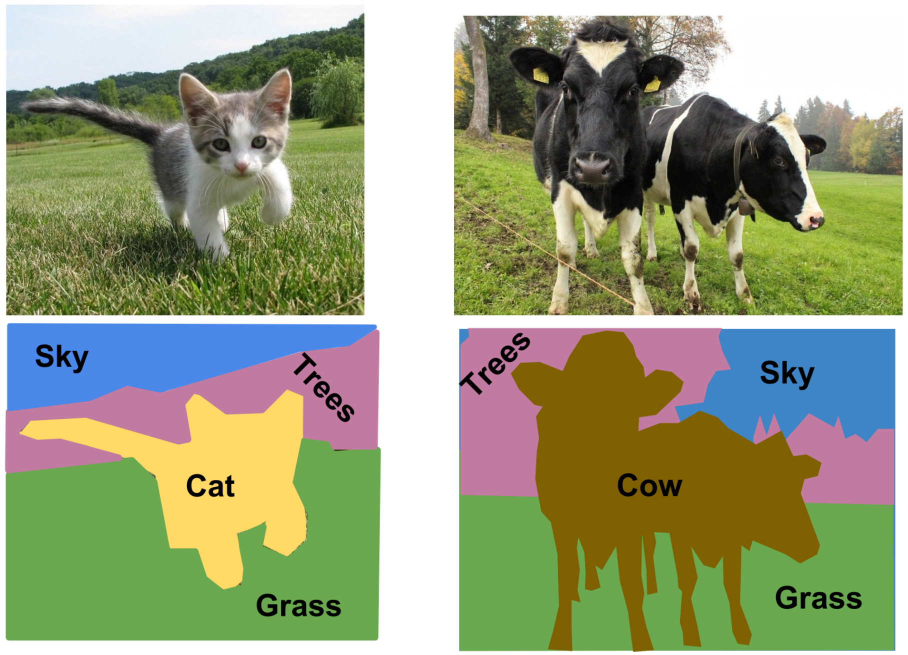
Source: [Mathworks]



Example of LIDAR-camera registration shown in Notebook 9!

Image Segmentation

- Semantic segmentation:
  - Label each pixel in the image with a category label
  - Doesn't differentiate instances, only cares about pixels



- Instance segmentation:
  - Label each pixel with its object instance
  - Identifies individual objects within each category

Source:
Stanford CS 231n lecture slides

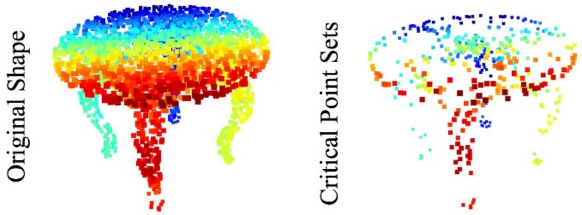
Information extraction and interpretation can also be done with LIDAR data!

Point cloud

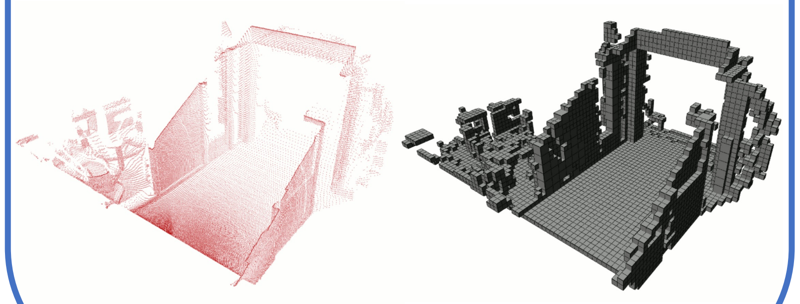
Images from PointNet and OctoMap

Features

Critical points

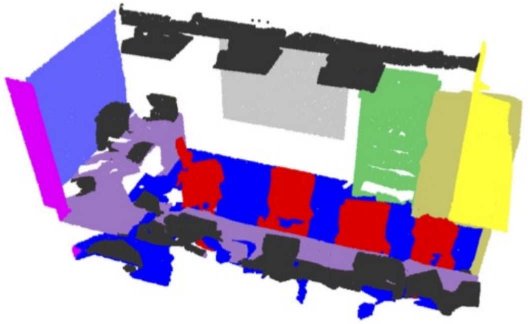


Occupancy grid map

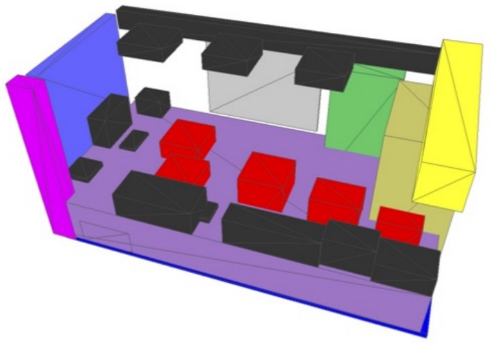


Tasks

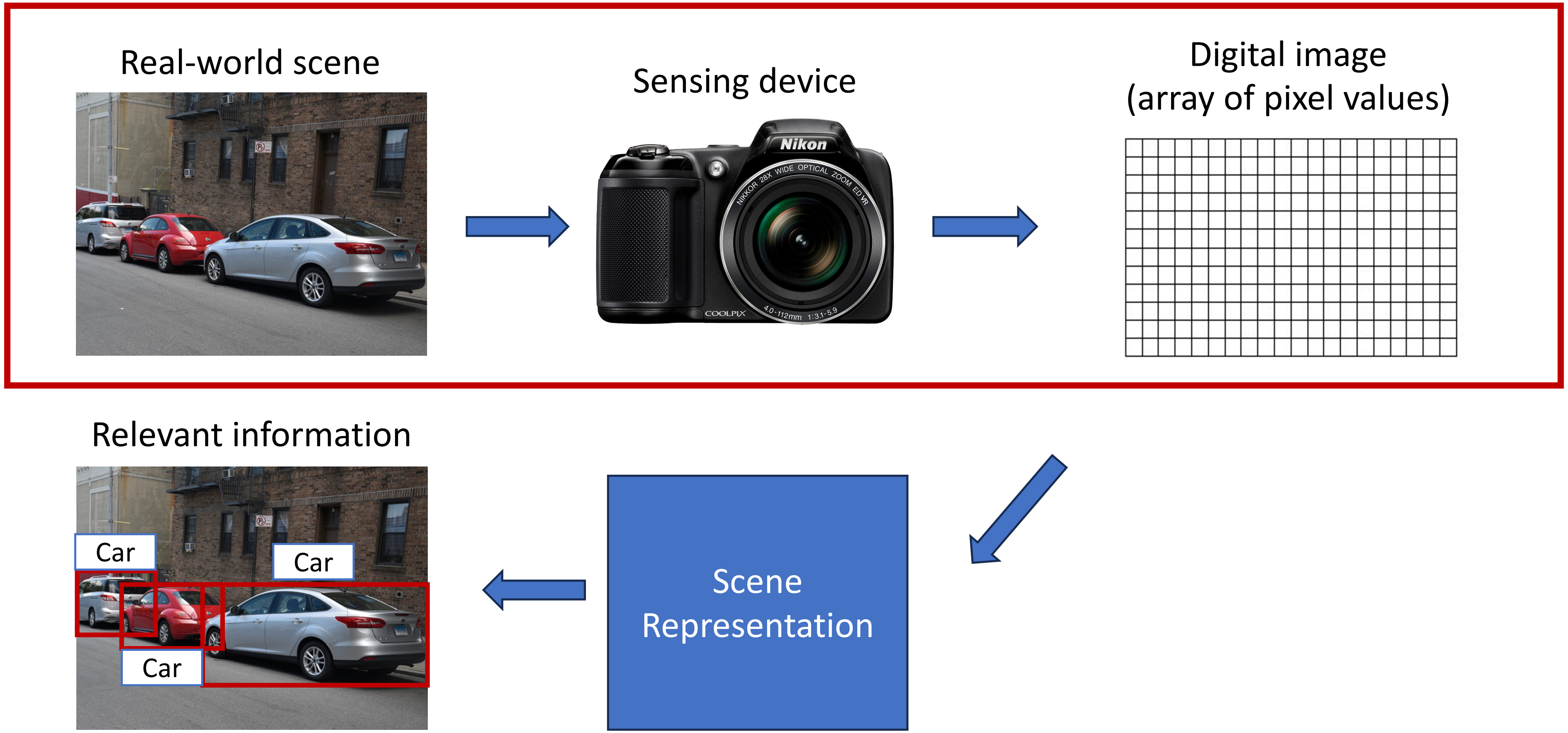
Semantic segmentation



Object detection

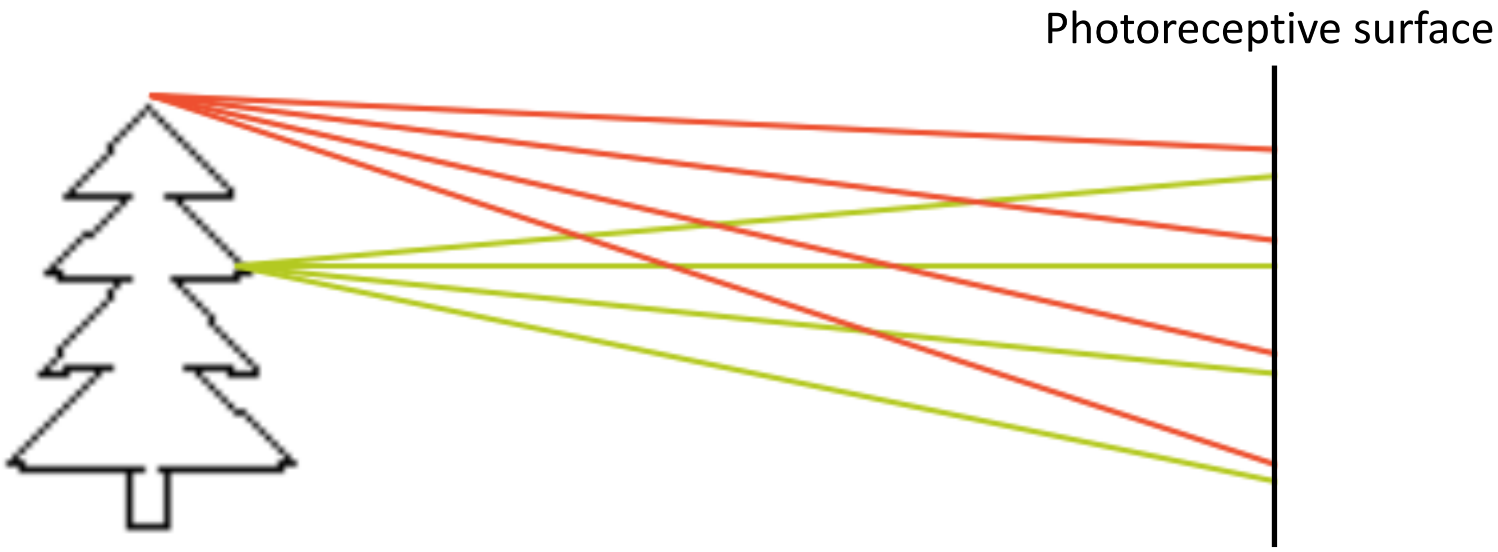


From Scenes to Digital Images



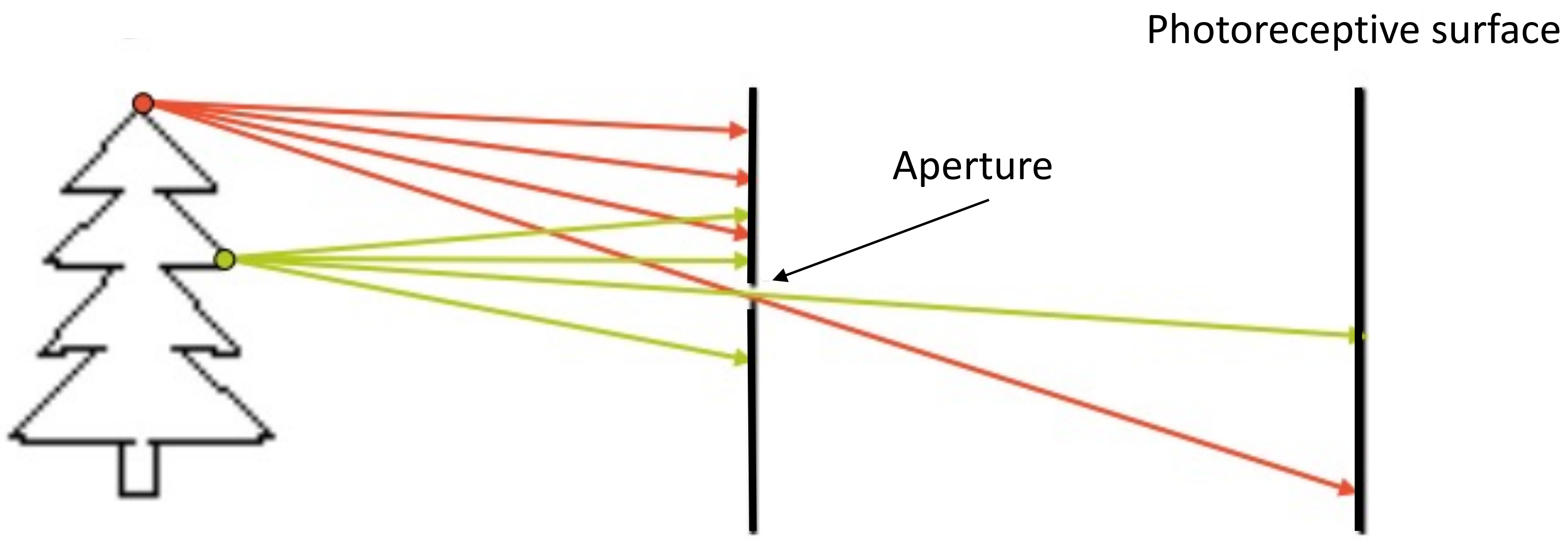
How to capture an image of the world?

- Light is reflected by the object and scattered in all directions
- If we simply add a photoreceptive surface, the captured image will be extremely blurred



Pinhole camera

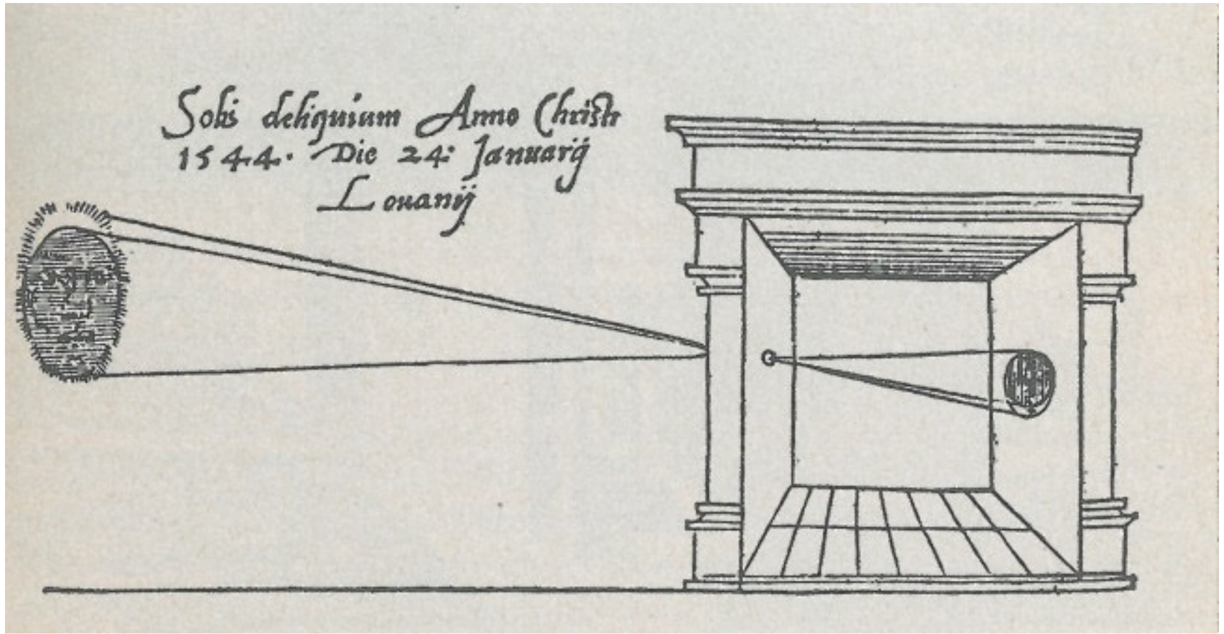
- Idea: add a barrier to block off most of the rays



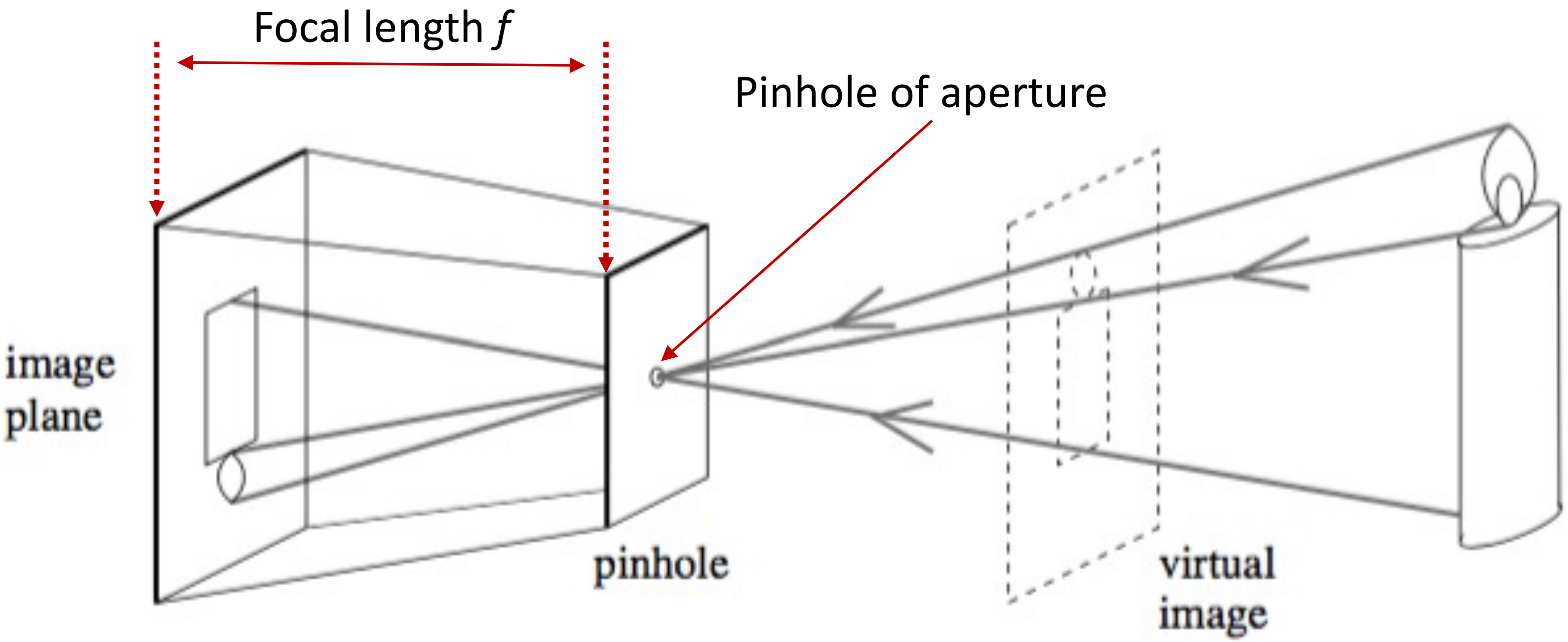
- Pinhole camera: a camera *without a lens* but with a tiny aperture, a *pinhole*

A long history

- Very old idea (several thousands of years BC)
- First clear description from Leonardo Da Vinci (1502)
- Oldest known published drawing of a camera obscura by Gemma Frisius (1544)



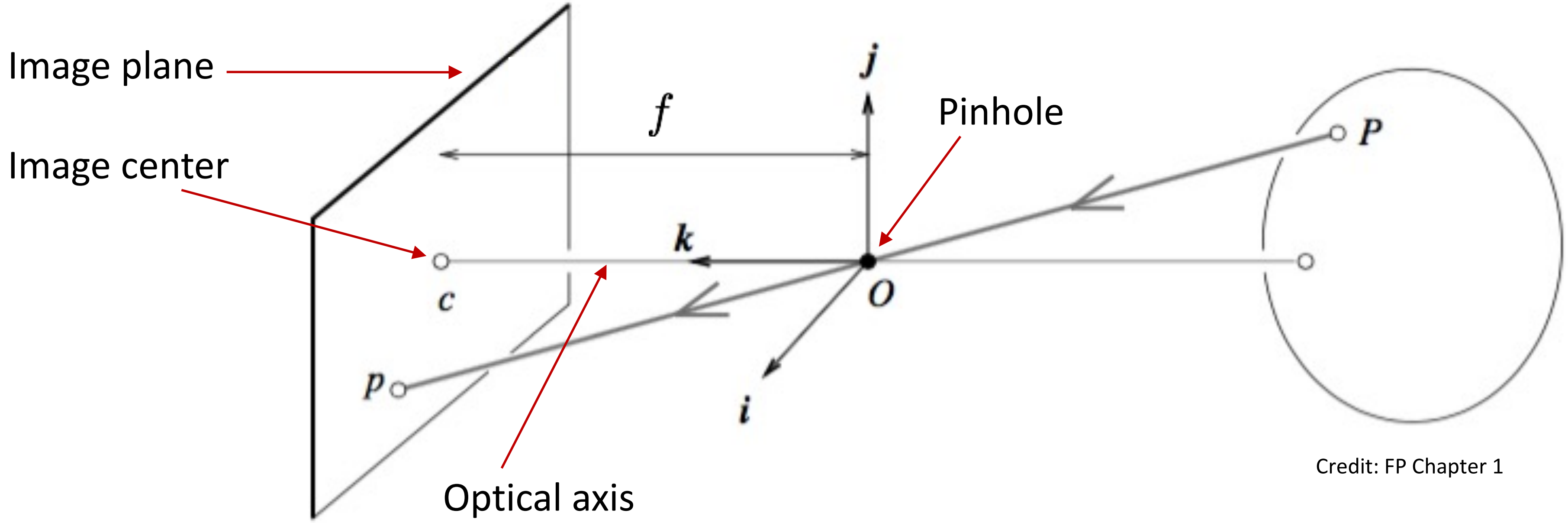
Pinhole camera



Credit: FP Chapter 1

- Perspective projection creates inverted images
- Sometimes it is convenient to consider a *virtual image* associated with a plane lying in front of the pinhole
- Virtual image not inverted but otherwise equivalent to the actual one

Pinhole perspective



$$P = (X, Y, Z)$$

Perspective

$$p = (x, y, z)$$

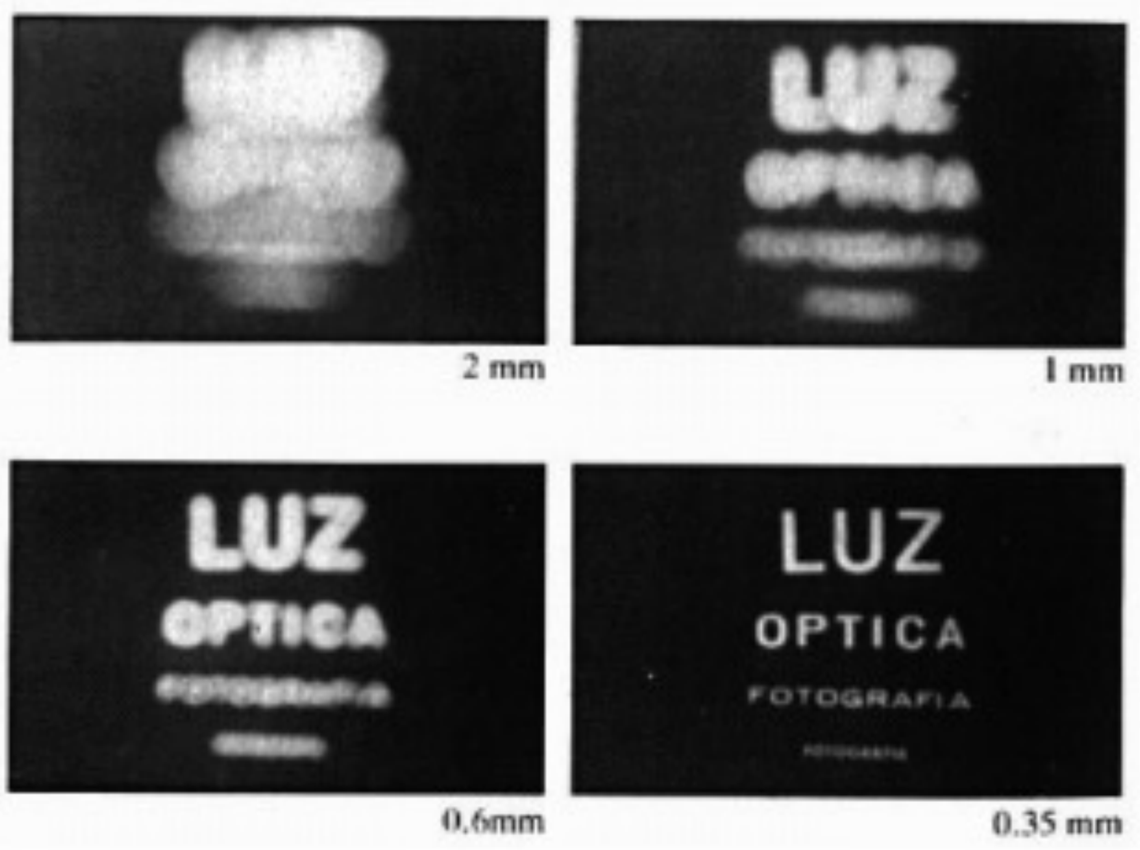
- Since *P*, *O*, and *p* are collinear: $\overline{Op} = \lambda \overline{OP}$ for some $\lambda \in R$
- Also, *z=f*, hence

$$\begin{cases} x = \lambda X \\ y = \lambda Y \\ z = \lambda Z \end{cases} \Leftrightarrow \lambda = \frac{x}{X} = \frac{y}{Y} = \frac{z}{Z} \quad \Rightarrow \quad \begin{cases} x = f\frac{X}{Z} \\ y = f\frac{Y}{Z} \end{cases}$$

Issues with pinhole camera

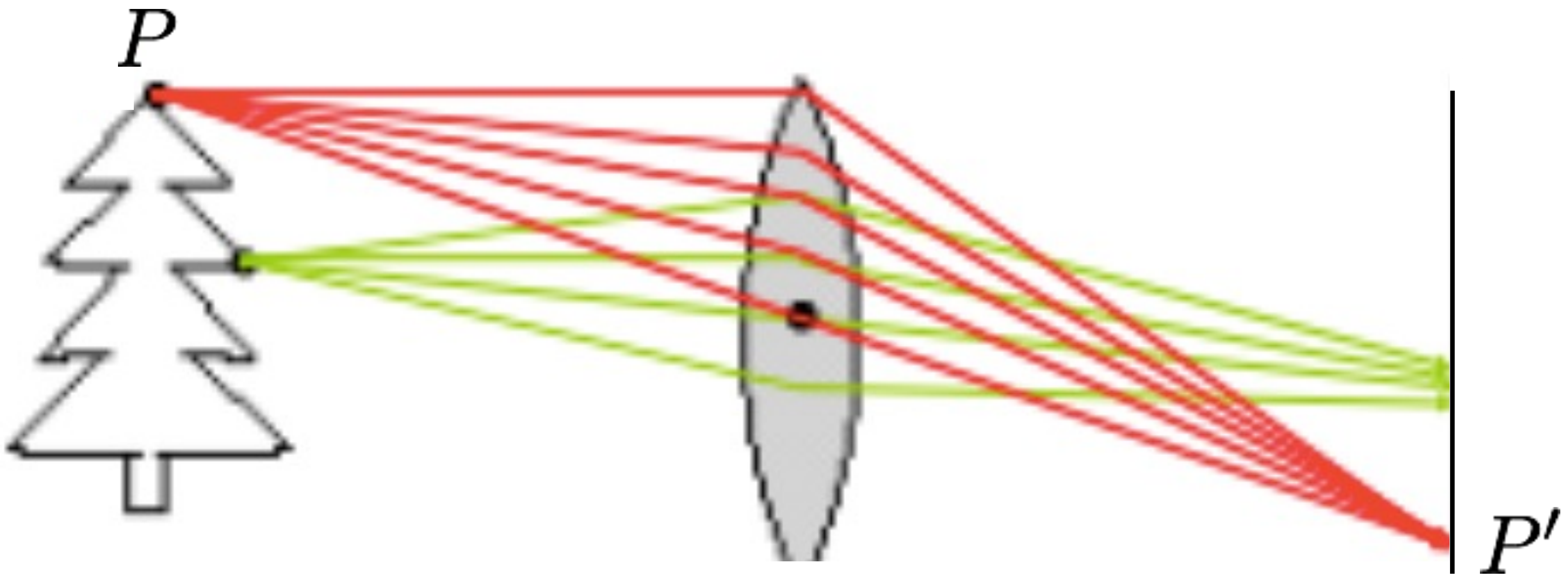
- Larger aperture -> greater number of light rays that pass through the aperture -> blur
- Smaller aperture -> fewer number of light rays that pass through the aperture -> darkness (+ diffraction)

- Solution: add a lens to replace the aperture!



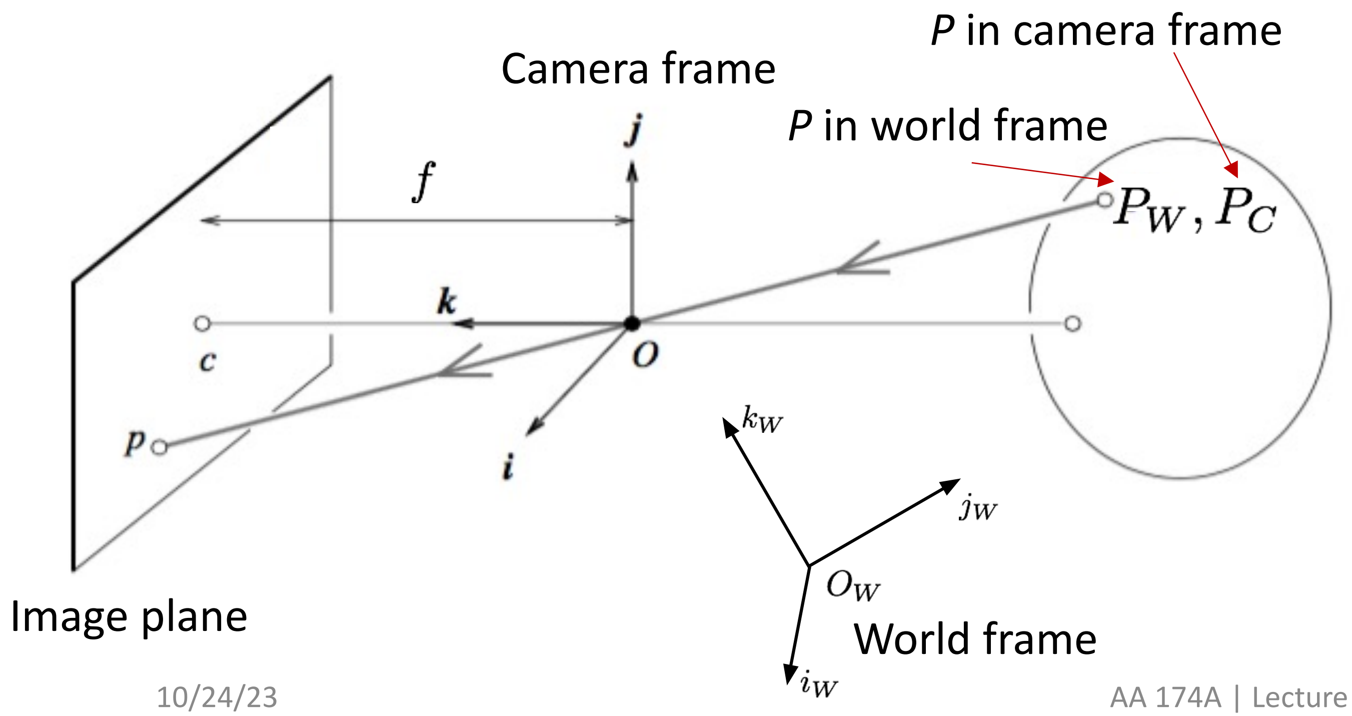
Lenses

- Lens: an optical element that focuses light by means of refraction



Perspective projection

- Goal: find how world points map in the camera image
- Assumption: pinhole camera model (*all results also hold under thin lens model, assuming camera is focused at ∞*)



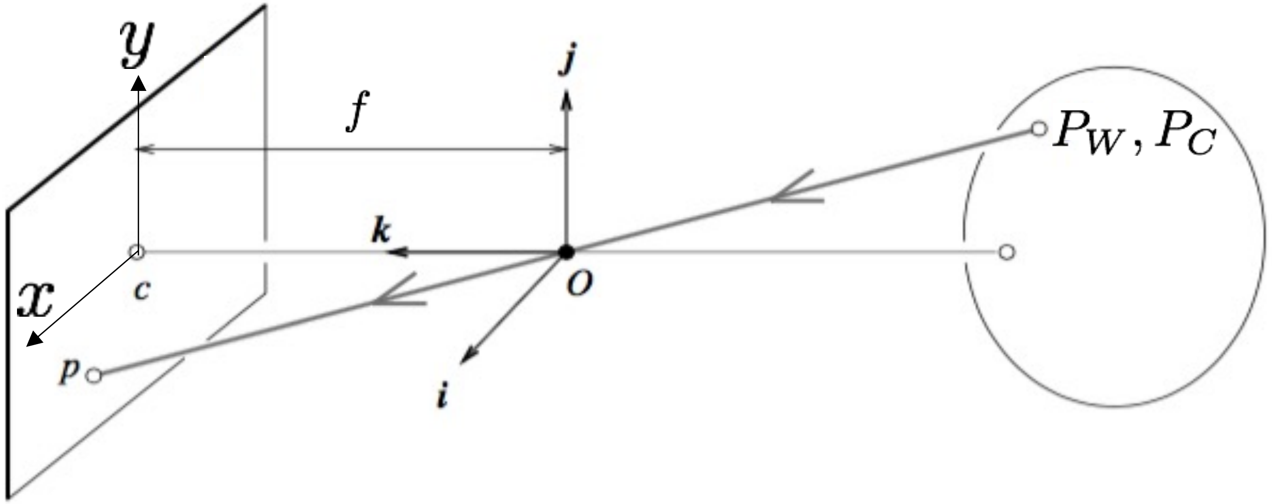
Roadmap:

1. Map P_c into *p* (image plane)
2. Map *p* into (u,v) (pixel coordinates)
3. Transform P_w into P_c

Step 1

- Task: Map $P_c = (X_C, Y_C, Z_C)$ into $p = (x, y)$ (image plane)
- From before

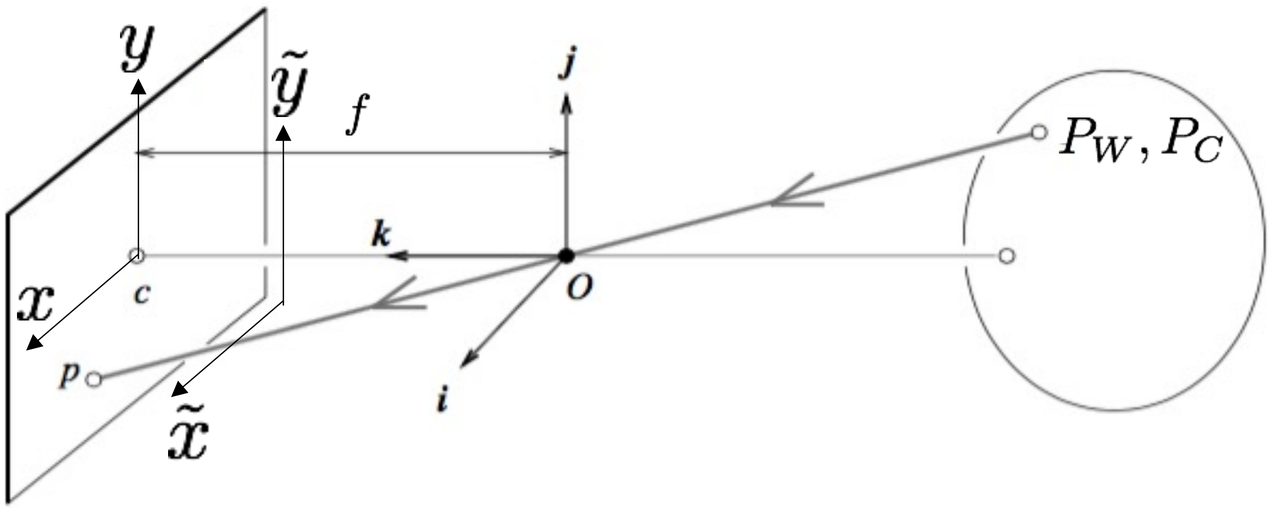
$$\begin{cases} x = f\frac{X_C}{Z_C} \\ y = f\frac{Y_C}{Z_C} \end{cases}$$

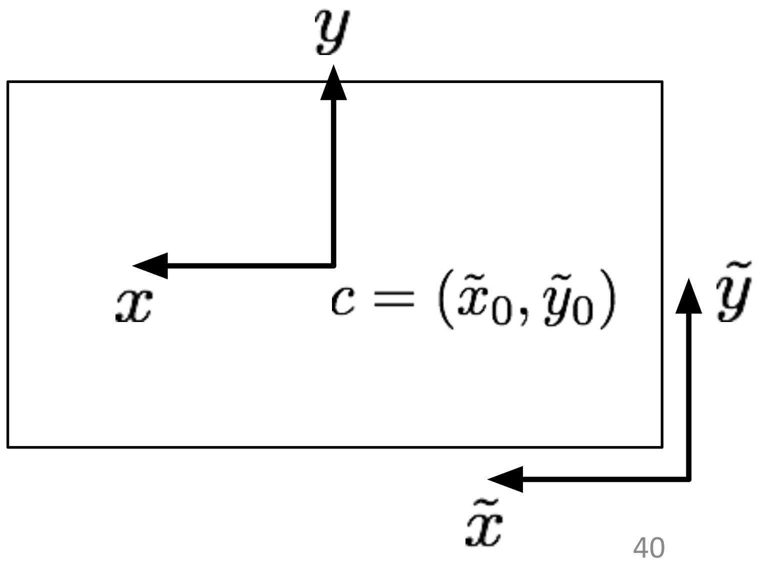


Step 2.a

- Actual origin of the camera coordinate system is usually at a corner (e.g., top left, bottom left)

$$\tilde{x} = f\,\frac{X_C}{Z_C} + \tilde{x}_0, \qquad \tilde{y} = f\,\frac{Y_C}{Z_C} + \tilde{y}_0,$$





Step 2.b

- Task: convert from image coordinates $(\tilde{x}, \tilde{y})$ to pixel coordinates (u, v)
- Let k_x and k_y be the number of pixels per unit distance in image coordinates in the *x* and *y* directions, respectively

$$u = k_x \tilde{x} = \overbrace{k_x\, f}^{\alpha} \frac{X_C}{Z_C} + \overbrace{k_x \tilde{x}_0}^{u_0}$$

$$v = k_y \tilde{y} = \underbrace{k_y f}_{\beta} \frac{Y_C}{Z_C} + \underbrace{k_y \tilde{y}_0}_{v_0}$$

$$\Rightarrow \quad \boxed{\begin{aligned} u &= \alpha \frac{X_C}{Z_C} + u_0 \\ v &= \beta \frac{Y_C}{Z_C} + v_0 \end{aligned}}$$

Nonlinear transformation

Homogeneous coordinates

- Goal: represent the transformation as a linear mapping
- Key idea: introduce homogeneous coordinates

Inhomogenous -> homogeneous

$$\begin{pmatrix} x \\ y \end{pmatrix} \Rightarrow \lambda \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \qquad \begin{pmatrix} x \\ y \\ z \end{pmatrix} \Rightarrow \lambda \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Homogenous -> inhomogeneous

$$\begin{pmatrix} x \\ y \\ w \end{pmatrix} \Rightarrow \begin{pmatrix} x/w \\ y/w \end{pmatrix} \qquad \begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix} \Rightarrow \begin{pmatrix} x/w \\ y/w \\ z/w \end{pmatrix}$$

Perspective projection in homogeneous coordinates

- Projection can be equivalently written in homogeneous coordinates

$$\overbrace{\begin{bmatrix} \alpha & 0 & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}}^{K}\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} = \begin{pmatrix} \alpha X_c + u_0 Z_c \\ \beta Y_c + v_0 Z_c \\ Z_c \end{pmatrix}$$

Camera matrix/
Matrix of intrinsic parameters

P_c in homogeneous coordinates

Homogeneous pixel coordinates

- In homogeneous coordinates, the mapping is linear:

$$p^h = [K \quad 0_{3\times 1}] P_C^h$$

Point p in homogeneous pixel coordinates

Point P_c in homogeneous camera coordinates

Skewness

- In some (rare) cases

Skew parameter

$$K = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

- When is $\gamma \neq 0$?
  - x- and y-axis of the camera are not perpendicular (unlikely)
  - For example, as a result of taking an image of an image
- Five parameters in total!

Next time: camera models & calibration