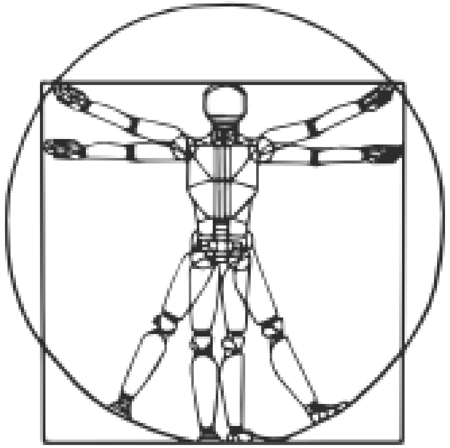
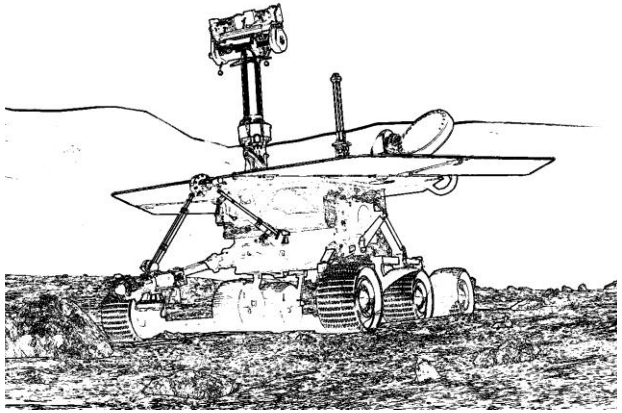




COMP417
Introduction to Robotics and Intelligent Systems

Lecture 4: Sensors and Actuators

Florian Shkurti

Computer Science Ph.D. student

florian@cim.mcgill.ca

McGill

Mobile Robotics Lab at McGill University

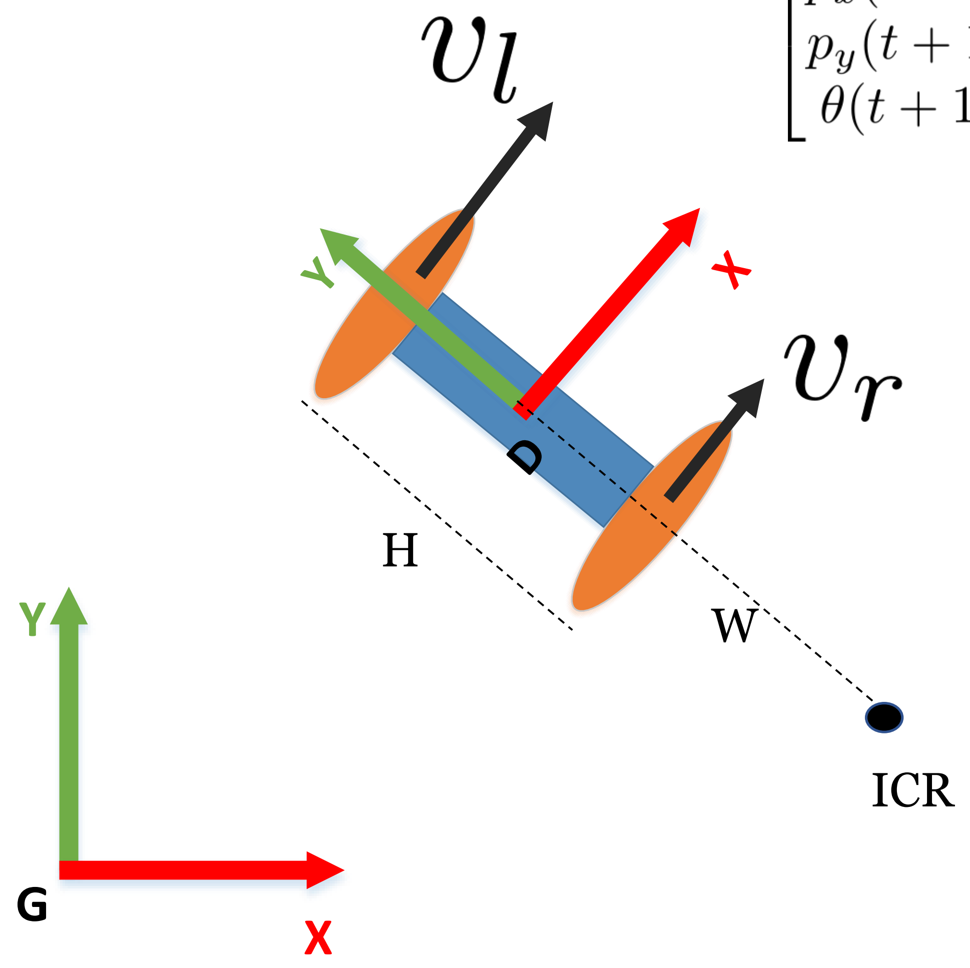
Quick recap from last week

- What is a non-holonomic constraint?

Noise

- Anything that we do not bother modelling with our model
- Example 1: “assume frictionless surface”
- Example 2: Taylor series expansion (only first few terms are dominant)
- With models, can be thought of as approximation error.

What type of vehicle is this?



$$\begin{bmatrix} p_x(t+1) \\ p_y(t+1) \\ \theta(t+1) \end{bmatrix} = \begin{bmatrix} \cos(\omega\delta t) & -\sin(\omega\delta t) & 0 \\ \sin(\omega\delta t) & \cos(\omega\delta t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x(t) - \mathrm{ICR}_x \\ p_y(t) - \mathrm{ICR}_y \\ \theta(t) \end{bmatrix} + \begin{bmatrix} \mathrm{ICR}_x \\ \mathrm{ICR}_y \\ \omega\delta t \end{bmatrix}$$

$$\mathrm{ICR} = [p_x - W\sin\theta, p_y + W\cos\theta]$$

Special cases:

- moving straight $v_l = v_r$
- in-place rotation $v_l = -v_r$
- rotation about the left wheel $v_l = 0$

The Dubins car is non-holonomic

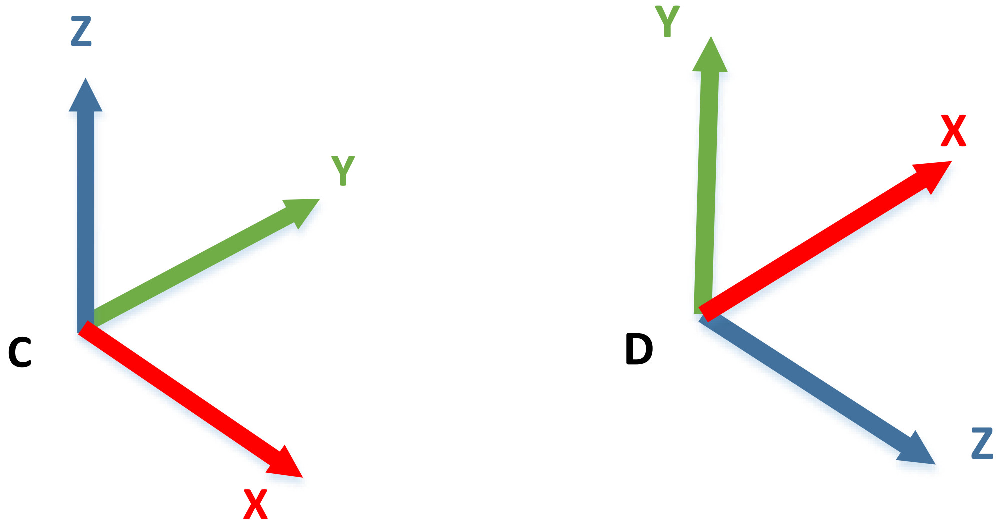
- Dubins car is constrained to move straight towards the direction it is currently heading. It cannot move sideways. It needs to "parallel park" to move laterally.

- In a small time interval dt the vehicle is going to move by δp_x and δp_y in the global frame of reference. Then from the dynamical system:

$$\begin{aligned} \delta p_x \sin(\theta) &= v_x \cos(\theta)\sin(\theta) dt \\ \delta p_y \cos(\theta) &= v_x \sin(\theta)\cos(\theta) dt \end{aligned} \Rightarrow \delta p_x \sin(\theta) = \delta p_y \cos(\theta) \Rightarrow v_x \sin(\theta) = v_y \cos(\theta)$$

Car is constrained to move along the line of current heading, i.e. non-holonomic

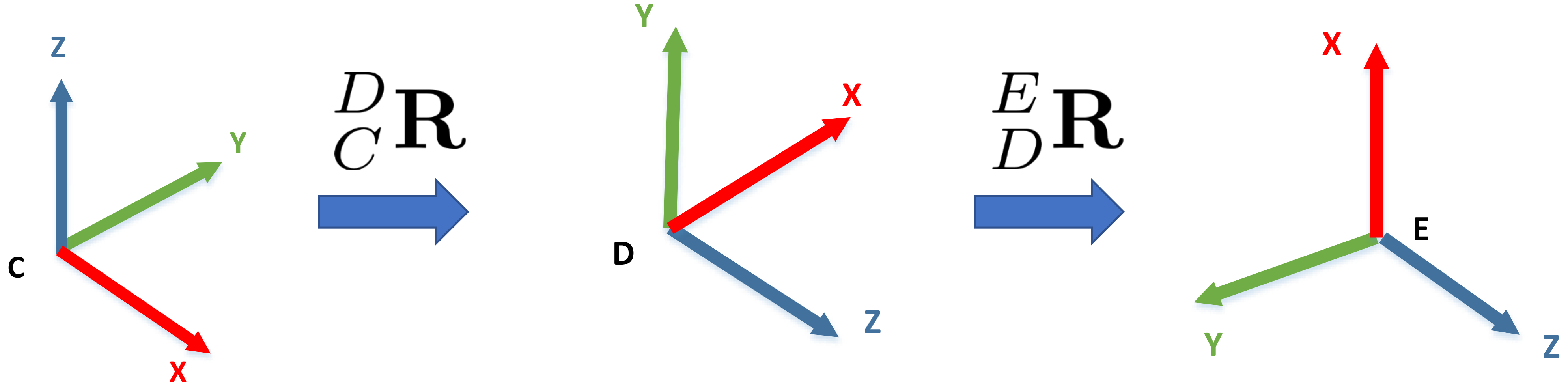
Example: finding a rotation matrix that rotates one vector to another



$$ {}^{D}_{C}\mathbf{R} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} $$

This matrix transforms the x-axis of frame C to the z-axis of frame D. Same for y and z axes.

Compound rotations



$$ {}^{E}_{C}\mathbf{R} = {}^{E}_{D}\mathbf{R}\,{}^{D}_{C}\mathbf{R} $$

Converting axis-angle to quaternion

- Given angle theta and axis v the equivalent quaternion representation is

$$\mathbf{q} = [\sin(\theta/2)v_1,\ \sin(\theta/2)v_2,\ \sin(\theta/2)v_3,\ \cos(\theta/2)]$$
$$\mathbf{q} = x\mathbf{i} + y\mathbf{j} + z\mathbf{k} + w$$

- Just like in the case of rotation matrices we denote the source and target frames of the rotation quaternion: ${}^{W}_{B}\mathbf{q}$

Today: Sensors and Actuators

- Our last introductory/background material lecture before we start talking about algorithms – the main topic of the course.

- Sensors:
  - Characteristics and types
  - Measurement noise
  - Required bandwidth

- Actuators:
  - Types of motors
  - Pulse-Width Modulation

Types of sensors

- General classification:
  - contact vs. non-contact
  - active vs. passive
  - sampling rate: fast vs. slow
  - local vs. non-local

- General examples:
  - vision
  - laser
  - radar
  - sonar
  - compass, gyroscope, accelerometer
  - touch (tactile)
  - infrared

Sensors

- Devices that can sense and measure physical properties of the environment.

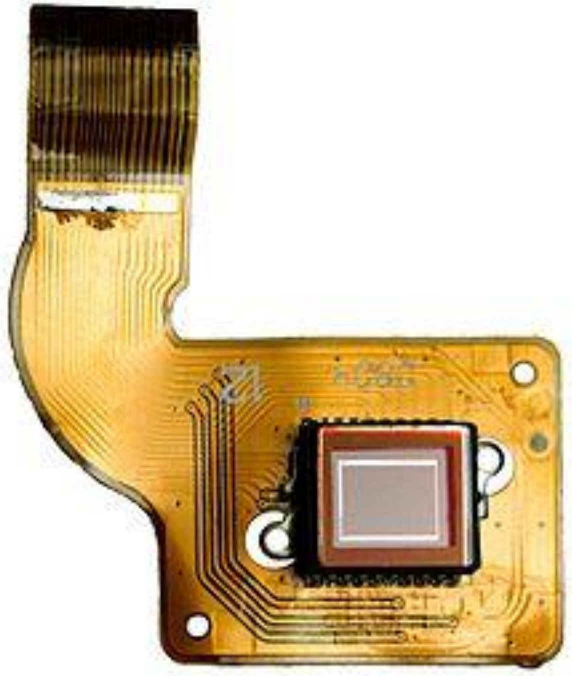
- Key phenomenon is **transduction** (conversion of energy from one form to another). E.g.:
  - Imaging sensors: light to pixel voltages
  - Depth sensors: mechanical pressure to voltage

- Measurements are **noisy**, and difficult to interpret

Sensors: general characteristics

- Sensitivity: (change of output) ÷ (change of input)
- Linearity: constancy of (output ÷ input)
- Measurement range: [min, max] or {min, max}
- Response time: time required for input change to cause output change
- Accuracy: difference between measurement and actual
- Repeatability/Drift: difference between repeated measures
- Resolution: smallest observable increment
- Bandwidth: required rate of data transfer
- SNR: signal-to-noise ratio

Sensors: vision



CCD image sensor

CCD (charge-coupled device) imaging sensors:

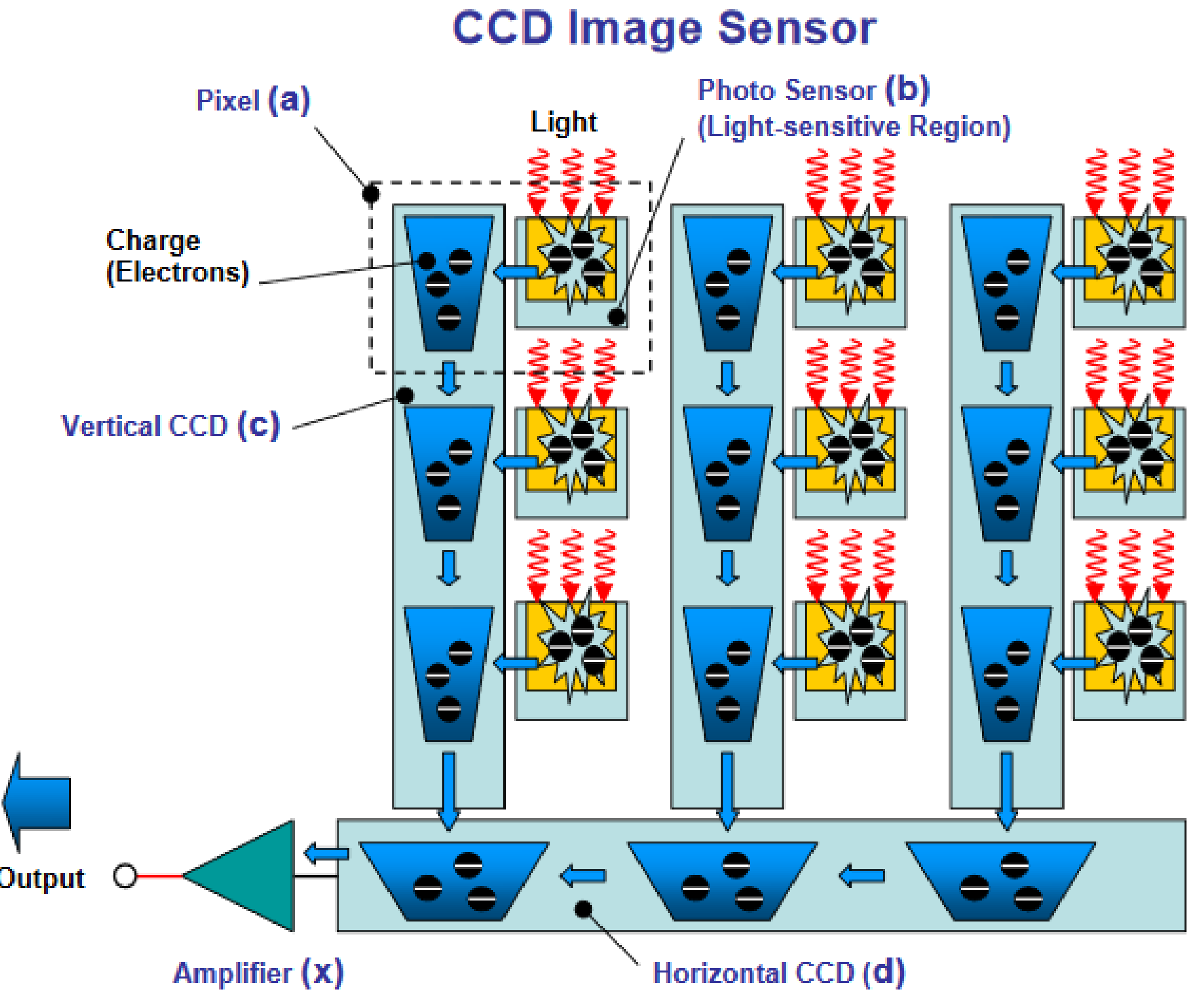
- Capacitor array accumulates electric charge proportional to light intensity.
- Each capacitor's charge is transferred to its neighbor.
- Last capacitor's charge gets amplified and output as voltage.
- (+) High-quality, low-noise images
- (-) Higher power consumption
- (-) Slow readout
- (-) Specialized fabrication

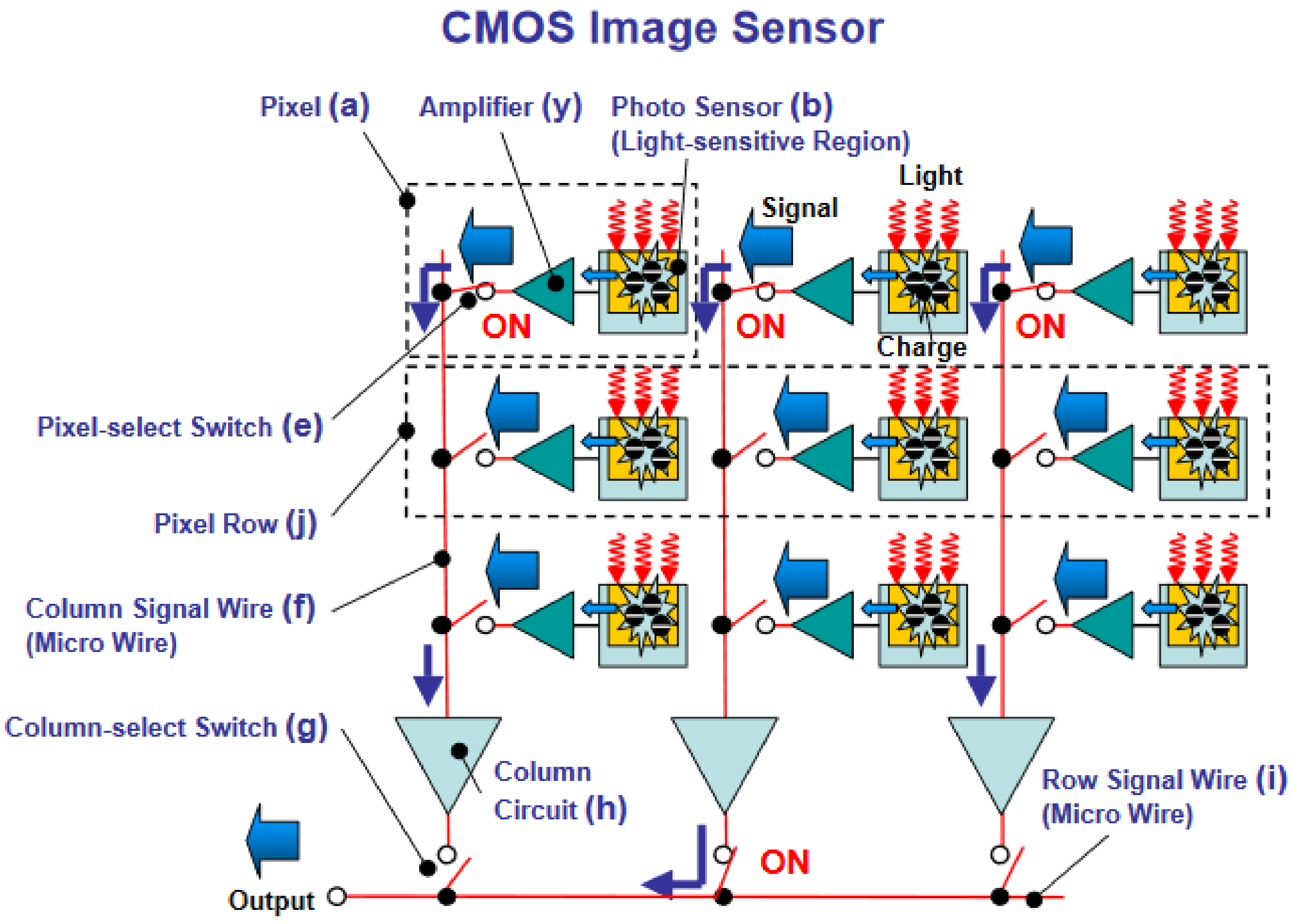
voltage → analog-to-digital converter → pixel value in {0, 255}

CMOS (complementary metal-oxide semi-conductor) imaging sensors:

- One amplifier per pixel
- (+) Low power
- (+) Fast readout
- (+) Easier to fabricate
- (-) Poor low-light sensitivity
- (-) Higher noise

CCD vs CMOS



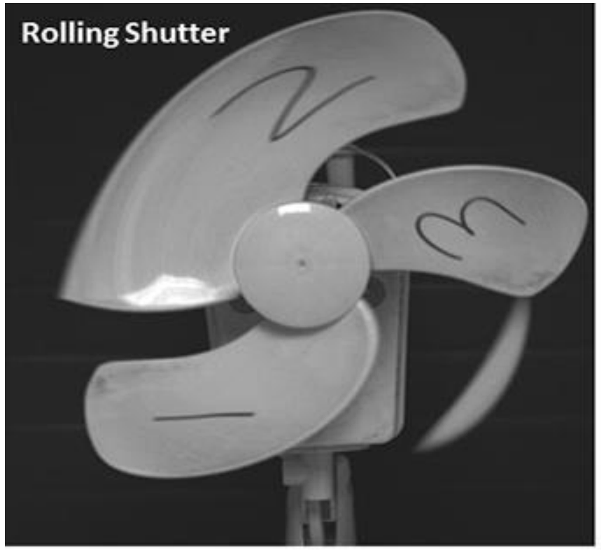


https://courses.cs.washington.edu/courses/cse467/08au/pdfs/lectures/07-cmos-ccd-imagers.pdf

Global vs. Rolling Shutter

Shutter = mechanism that allows light to hit the imaging sensor

Shutter “speed” = Exposure time = time duration in which the sensor is exposed to light



Rolling shutter

Reading RGB images from a camera

Each pixel contains an intensity value from 0...255

600 x 1000 pixels

600 x 1000 pixels

600 x 1000 pixels

Reading images from a camera

A matrix of
600 x 1000 x 3 =
~ 1.8 million numbers

Each pixel contains an intensity value from 0…255

08	02	22	97	38	15	00	40	00	75	04	05	07	78	52	12	50	77	91	08
49	49	99	40	17	81	18	57	60	87	17	40	98	43	69	48	04	56	62	00
81	49	31	73	55	79	14	29	93	71	40	67	53	88	30	03	49	13	36	65
52	70	95	23	04	60	11	42	69	24	68	56	01	32	56	71	37	02	36	91
22	31	16	71	51	67	63	89	41	92	36	54	22	40	40	28	66	33	13	80
24	47	32	60	99	03	45	02	44	75	33	53	78	36	84	20	35	17	12	50
32	98	81	28	64	23	67	10	26	38	40	67	59	54	70	66	18	38	64	70
67	26	20	68	02	62	12	20	95	63	94	39	63	08	40	91	66	49	94	21
24	55	58	05	66	73	99	26	97	17	78	78	96	83	14	88	34	89	63	72
21	36	23	09	75	00	76	44	20	45	35	14	00	61	33	97	34	31	33	95
78	17	53	28	22	75	31	67	15	94	03	80	04	62	16	14	09	53	56	92
16	39	05	42	96	35	31	47	55	58	88	24	00	17	54	24	36	29	85	57
86	56	00	48	35	71	89	07	05	44	44	37	44	60	21	58	51	54	17	58
19	80	81	68	05	94	47	69	28	73	92	13	86	52	17	77	04	89	55	40
04	52	08	83	97	35	99	16	07	97	57	32	16	26	26	79	33	27	98	66
88	36	68	87	57	62	20	72	03	46	33	67	46	55	12	32	63	93	53	69
04	42	16	73	38	25	39	11	24	94	72	18	08	46	29	32	40	62	76	36
20	69	36	41	72	30	23	88	34	62	99	69	82	67	59	85	74	04	36	16
20	73	35	29	78	31	90	01	74	31	49	71	48	86	81	16	23	57	05	54
01	70	54	71	83	51	54	69	16	92	33	48	61	43	52	01	89	19	67	48

600 x 1000 pixels

08	02	22	97	38	15	00	40	00	75	04	05	07	78	52	12	50	77	91	08
49	49	99	40	17	81	18	57	60	87	17	40	98	43	69	48	04	56	62	00
81	49	31	73	55	79	14	29	93	71	40	67	53	88	30	03	49	13	36	65
52	70	95	23	04	60	11	42	69	24	68	56	01	32	56	71	37	02	36	91
22	31	16	71	51	67	63	89	41	92	36	54	22	40	40	28	66	33	13	80
24	47	32	60	99	03	45	02	44	75	33	53	78	36	84	20	35	17	12	50
32	98	81	28	64	23	67	10	26	38	40	67	59	54	70	66	18	38	64	70
67	26	20	68	02	62	12	20	95	63	94	39	63	08	40	91	66	49	94	21
24	55	58	05	66	73	99	26	97	17	78	78	96	83	14	88	34	89	63	72
21	36	23	09	75	00	76	44	20	45	35	14	00	61	33	97	34	31	33	95
78	17	53	28	22	75	31	67	15	94	03	80	04	62	16	14	09	53	56	92
16	39	05	42	96	35	31	47	55	58	88	24	00	17	54	24	36	29	85	57
86	56	00	48	35	71	89	07	05	44	44	37	44	60	21	58	51	54	17	58
19	80	81	68	05	94	47	69	28	73	92	13	86	52	17	77	04	89	55	40
04	52	08	83	97	35	99	16	07	97	57	32	16	26	26	79	33	27	98	66
88	36	68	87	57	62	20	72	03	46	33	67	46	55	12	32	63	93	53	69
04	42	16	73	38	25	39	11	24	94	72	18	08	46	29	32	40	62	76	36
20	69	36	41	72	30	23	88	34	62	99	69	82	67	59	85	74	04	36	16
20	73	35	29	78	31	90	01	74	31	49	71	48	86	81	16	23	57	05	54
01	70	54	71	83	51	54	69	16	92	33	48	61	43	52	01	89	19	67	48

600 x 1000 pixels

08	02	22	97	38	15	00	40	00	75	04	05	07	78	52	12	50	77	91	08
49	49	99	40	17	81	18	57	60	87	17	40	98	43	69	48	04	56	62	00
81	49	31	73	55	79	14	29	93	71	40	67	53	88	30	03	49	13	36	65
52	70	95	23	04	60	11	42	69	24	68	56	01	32	56	71	37	02	36	91
22	31	16	71	51	67	63	89	41	92	36	54	22	40	40	28	66	33	13	80
24	47	32	60	99	03	45	02	44	75	33	53	78	36	84	20	35	17	12	50
32	98	81	28	64	23	67	10	26	38	40	67	59	54	70	66	18	38	64	70
67	26	20	68	02	62	12	20	95	63	94	39	63	08	40	91	66	49	94	21
24	55	58	05	66	73	99	26	97	17	78	78	96	83	14	88	34	89	63	72
21	36	23	09	75	00	76	44	20	45	35	14	00	61	33	97	34	31	33	95
78	17	53	28	22	75	31	67	15	94	03	80	04	62	16	14	09	53	56	92
16	39	05	42	96	35	31	47	55	58	88	24	00	17	54	24	36	29	85	57
86	56	00	48	35	71	89	07	05	44	44	37	44	60	21	58	51	54	17	58
19	80	81	68	05	94	47	69	28	73	92	13	86	52	17	77	04	89	55	40
04	52	08	83	97	35	99	16	07	97	57	32	16	26	26	79	33	27	98	66
88	36	68	87	57	62	20	72	03	46	33	67	46	55	12	32	63	93	53	69
04	42	16	73	38	25	39	11	24	94	72	18	08	46	29	32	40	62	76	36
20	69	36	41	72	30	23	88	34	62	99	69	82	67	59	85	74	04	36	16
20	73	35	29	78	31	90	01	74	31	49	71	48	86	81	16	23	57	05	54
01	70	54	71	83	51	54	69	16	92	33	48	61	43	52	01	89	19	67	48

600 x 1000 pixels

Computer/robot vision

08	02	22	97	38	15	00	40	00	75	04	05	07	78	52	12	50	77	91	08
49	49	99	40	17	81	18	57	60	87	17	40	98	43	69	48	04	56	62	00
81	49	31	73	55	79	14	29	93	71	40	67	53	88	30	03	49	13	36	65
52	70	95	23	04	60	11	42	69	24	68	56	01	32	56	71	37	02	36	91
22	31	16	71	51	67	63	89	41	92	36	54	22	40	40	28	66	33	13	80
24	47	32	60	99	03	45	02	44	75	33	53	78	36	84	20	35	17	12	50
32	98	81	28	64	23	67	10	26	38	40	67	59	54	70	66	18	38	64	70
67	26	20	68	02	62	12	20	95	63	94	39	63	08	40	91	66	49	94	21
24	55	58	05	66	73	99	26	97	17	78	78	96	83	14	88	34	89	63	72
21	36	23	09	75	00	76	44	20	45	35	14	00	61	33	97	34	31	33	95
78	17	53	28	22	75	31	67	15	94	03	80	04	62	16	14	09	53	56	92
16	39	05	42	96	35	31	47	55	58	88	24	00	17	54	24	36	29	85	57
86	56	00	48	35	71	89	07	05	44	44	37	44	60	21	58	51	54	17	58
19	80	81	68	05	94	47	69	28	73	92	13	86	52	17	77	04	89	55	40
04	52	08	83	97	35	99	16	07	97	57	32	16	26	26	79	33	27	98	66
88	36	68	87	57	62	20	72	03	46	33	67	46	55	12	32	63	93	53	69
04	42	16	73	38	25	39	11	24	94	72	18	08	46	29	32	40	62	76	36
20	69	36	41	72	30	23	88	34	62	99	69	82	67	59	85	74	04	36	16
20	73	35	29	78	31	90	01	74	31	49	71	48	86	81	16	23	57	05	54
01	70	54	71	83	51	54	69	16	92	33	48	61	43	52	01	89	19	67	48

Structured numbers

?

1. I’m seeing a parrot
2. I’m seeing a toy bicycle
3. The parrot is riding the bicycle
4. The bicycle is on top of a desk
5. Is this physically plausible?
6. Where is the parrot in 3D w.r.t. the camera?
7. Where will the parrot go next?
8. What is the speed of the parrot?

Conclusions/Inference/Deduction/Estimation

Computer/robot vision

08	02	22	97	38	15	00	40	00	75	04	05	07	78	52	12	50	77	91	08
49	49	99	40	17	81	18	57	60	87	17	40	98	43	69	48	04	56	62	00
81	49	31	73	55	79	14	29	93	71	40	67	53	88	30	03	49	13	36	65
52	70	95	23	04	60	11	42	69	24	68	56	01	32	56	71	37	02	36	91
22	31	16	71	51	67	63	89	41	92	36	54	22	40	40	28	66	33	13	80
24	47	32	60	99	03	45	02	44	75	33	53	78	36	84	20	35	17	12	50
32	98	81	28	64	23	67	10	26	38	40	67	59	54	70	66	18	38	64	70
67	26	20	68	02	62	12	20	95	63	94	39	63	08	40	91	66	49	94	21
24	55	58	05	66	73	99	26	97	17	78	78	96	83	14	88	34	89	63	72
21	36	23	09	75	00	76	44	20	45	35	14	00	61	33	97	34	31	33	95
78	17	53	28	22	75	31	67	15	94	03	80	04	62	16	14	09	53	56	92
16	39	05	42	96	35	31	47	55	58	88	24	00	17	54	24	36	29	85	57
86	56	00	48	35	71	89	07	05	44	44	37	44	60	21	58	51	54	17	58
19	80	81	68	05	94	47	69	28	73	92	13	86	52	17	77	04	89	55	40
04	52	08	83	97	35	99	16	07	97	57	32	16	26	26	79	33	27	98	66
88	36	68	87	57	62	20	72	03	46	33	67	46	55	12	32	63	93	53	69
04	42	16	73	38	25	39	11	24	94	72	18	08	46	29	32	40	62	76	36
20	69	36	41	72	30	23	88	34	62	99	69	82	67	59	85	74	04	36	16
20	73	35	29	78	31	90	01	74	31	49	71	48	86	81	16	23	57	05	54
01	70	54	71	83	51	54	69	16	92	33	48	61	43	52	01	89	19	67	48

Structured numbers

?

1. I'm seeing a parrot
2. I'm seeing a toy bicycle
3. The parrot is riding the bicycle
4. The bicycle is on top of a desk
5. Is this physically plausible?
6. Where is the parrot in 3D w.r.t. the camera?
7. Where will the parrot go next?
8. What is the speed of the parrot?

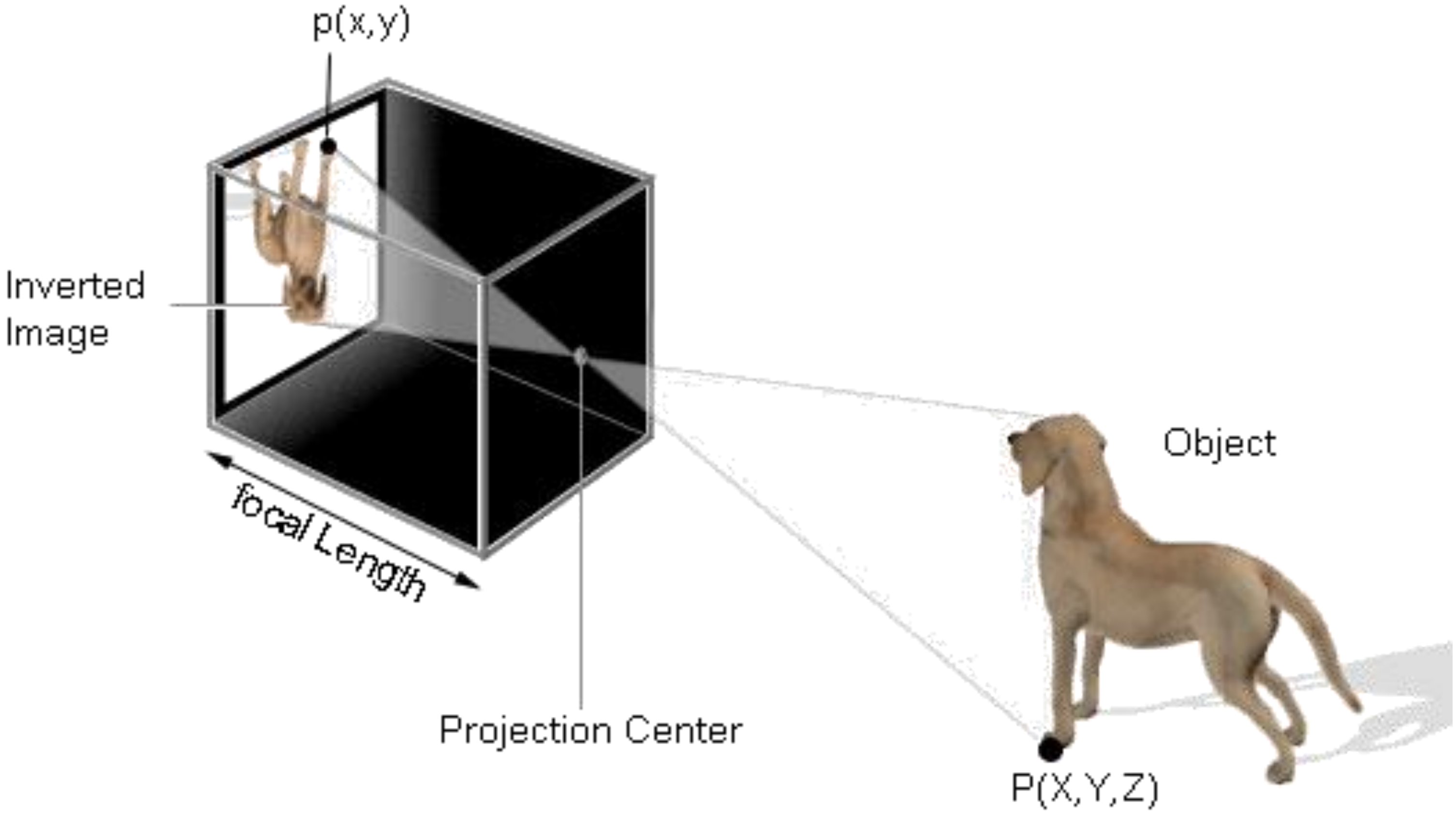
Conclusions/Inference/Deduction/Estimation

Camera lenses

- Lens determines:
  - image distortion
  - focus
  - sharpness or blur

- Lens characteristics:
  - focal length
  - aperture
  - depth-of-field

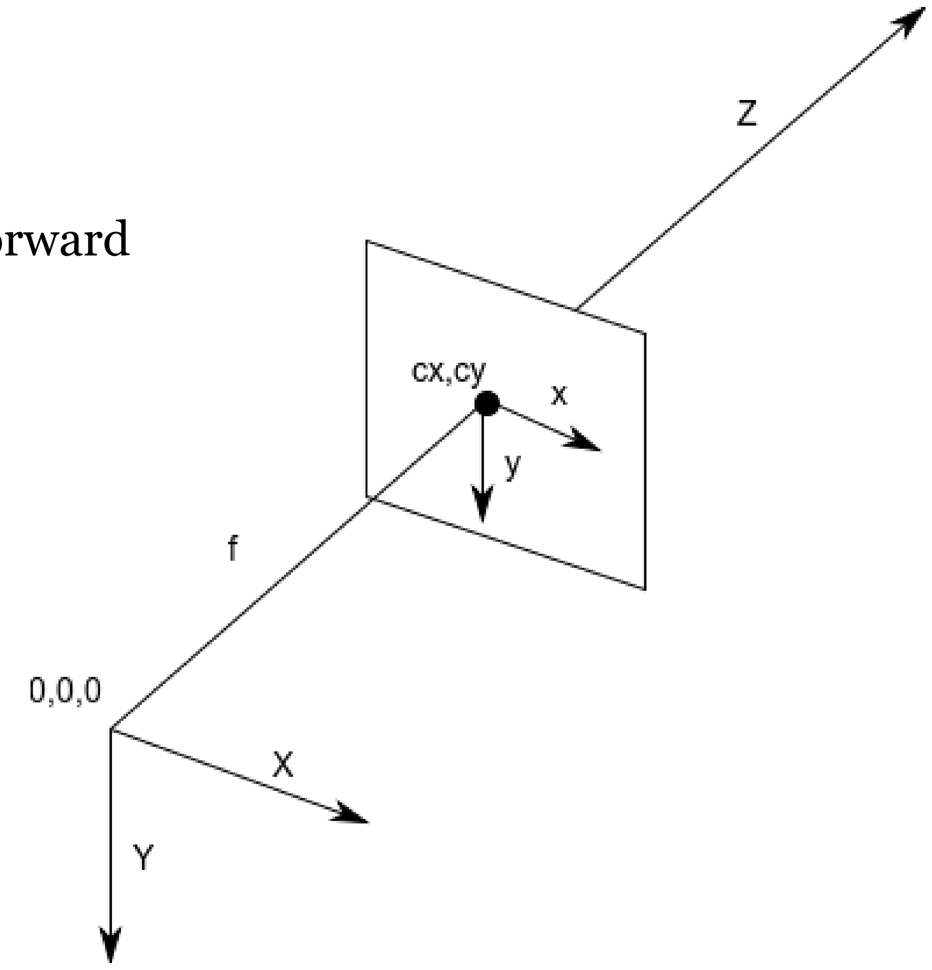
Pinhole Camera Model



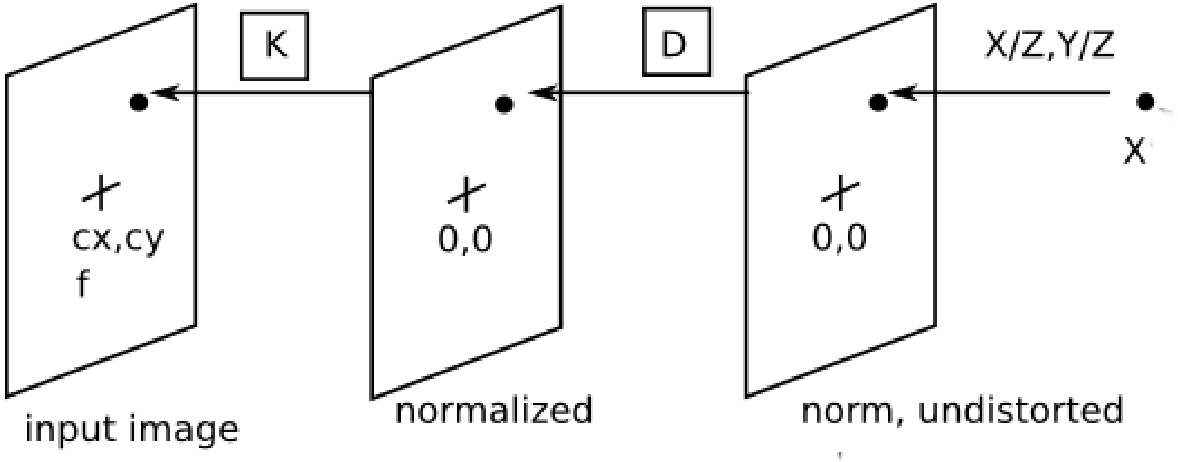
We know **approximately** how a 3D point (X,Y,Z) projects to pixel (x,y)
We call this the ***pinhole projection model***

Pinhole Camera:
Avoiding to think about image inversion

We push the image plane forward
along the positive z-axis

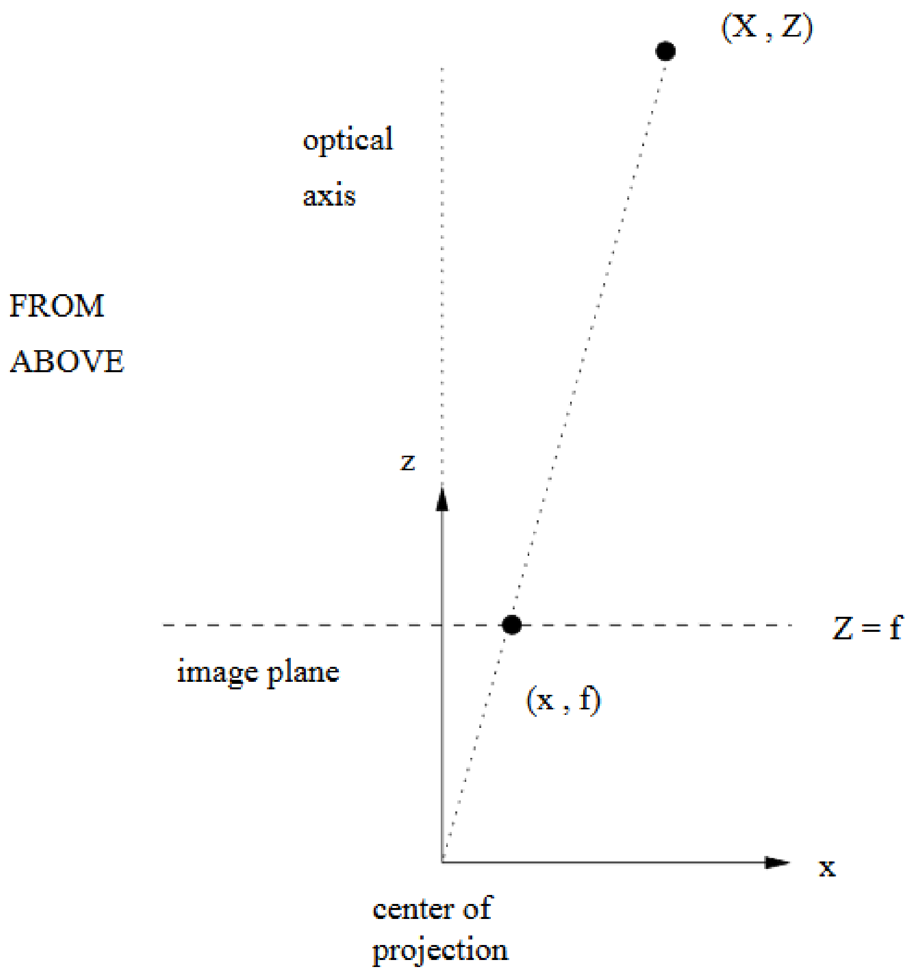


Pinhole Camera: From 3D points to pixels



(1) $[x,y] = \pi(X,Y,Z)$ perspective projection
(2) $[x^*, y^*] = D(x,y)$ lens distortion
(3) $[u,v,1] = K[x^*,y^*,1]'$ pixel coordinates

(1) Perspective projection
$[x,y] = \pi(X,Y,Z)$



http://www.cim.mcgill.ca/%7Elanger/558.html

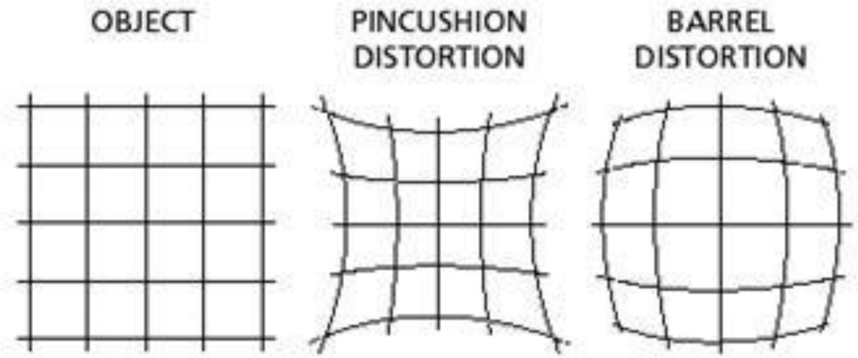
By similar triangles: $x/f = X/Z$

So, $x = f * X/Z$ and similarly $y = f * Y/Z$

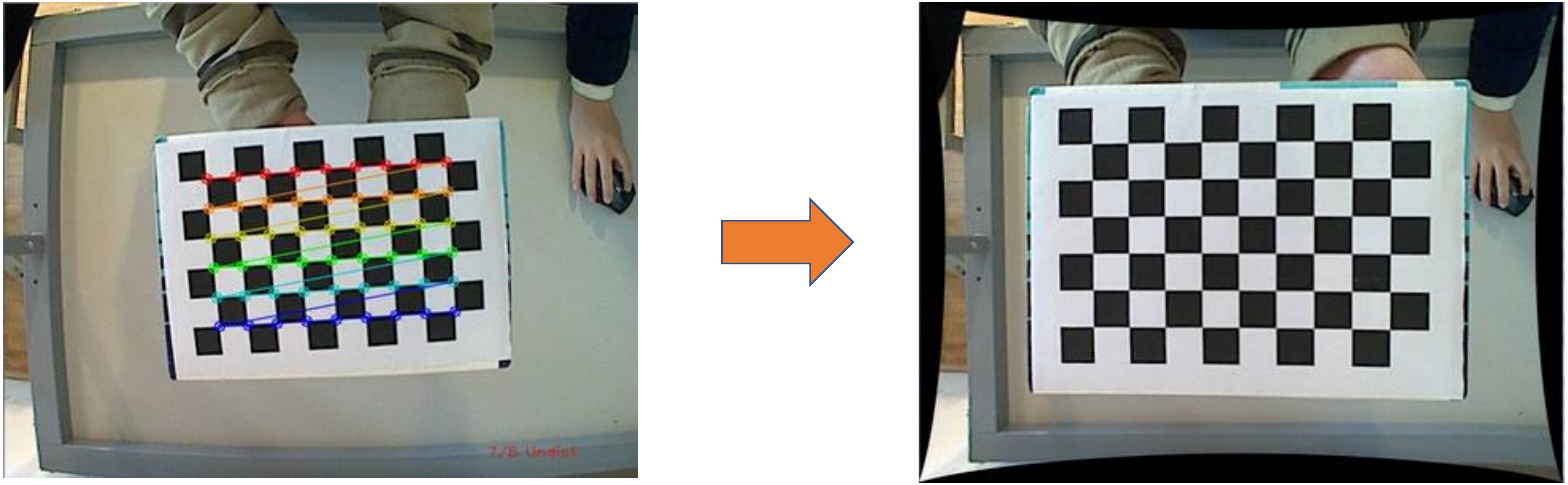
Problem: we just lost depth (Z) information by doing this projection, i.e. depth is now uncertain.

(2) Lens distortion

$$[x^*, y^*] = D(x,y)$$



(2) Estimating parameters of lens distortion: [x*, y*] = D(x,y)



$$x^* = x\frac{1 + k_1 r^2 + k_2 r^4 + k_3 r^6}{1 + k_4 r^2 + k_5 r^4 + k_6 r^6} \quad \text{where} \quad r = x^2 + y^2$$

(3) From metric to pixel coordinates [u,v,1] = K[x*,y*,1]

- Let f_x and f_y describe the number of pixels per mm in the sensor

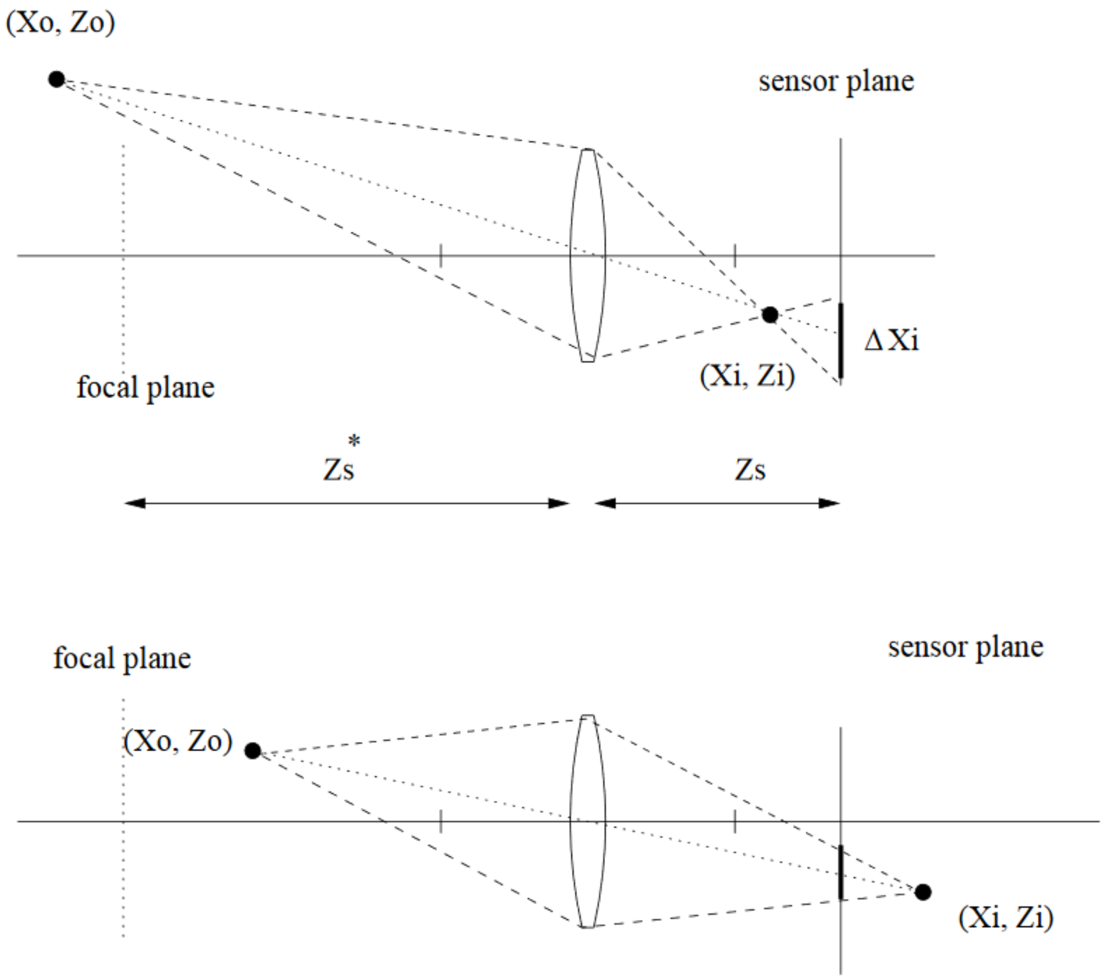
- Typically (0,0) is the pixel coordinate of the top left pixel, not of the principal point.

- Let c_x and c_y describe the pixel coordinates of the principal point.

- Then $\begin{aligned} u &= f_x x^* + c_x \\ v &= f_y y^* + c_y \end{aligned}$ and $K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$.

The unknown parameters in steps 2 and 3 are estimated through camera calibration by observing the chessboard from multiple viewpoints.

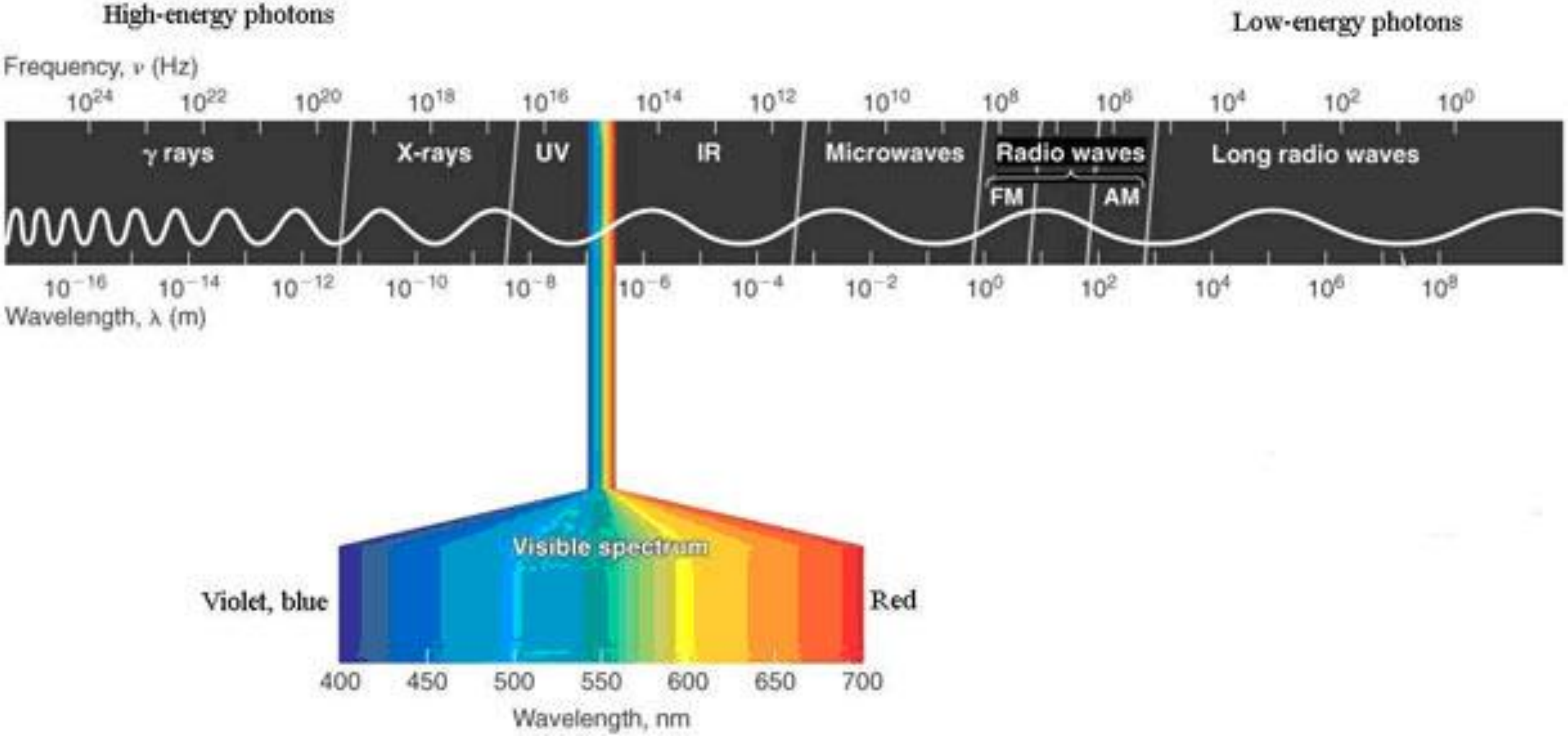
Non-pinhole cameras: thin lens model



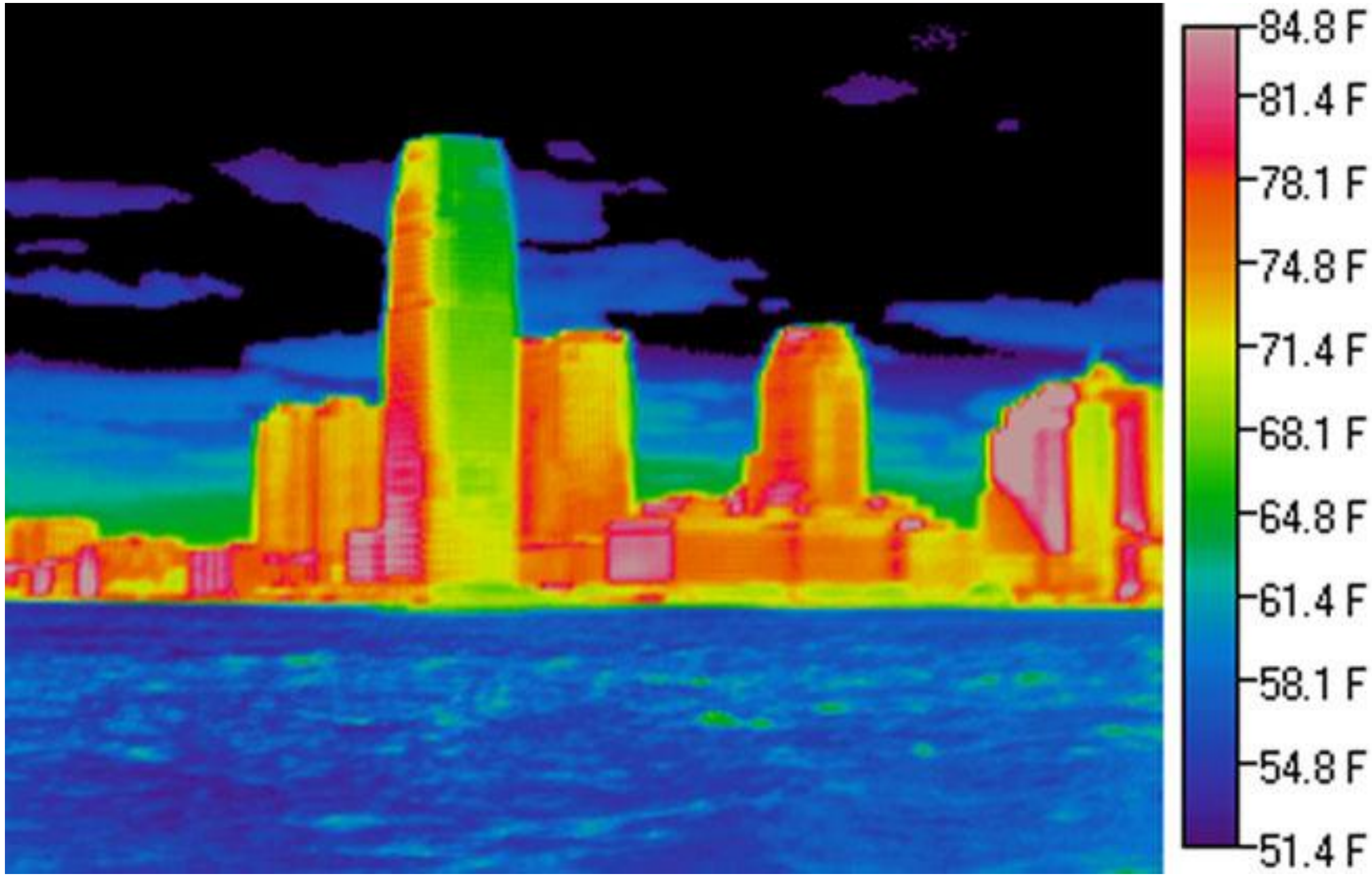
Unlike the pinhole camera, this is able to model blur.

http://www.cim.mcgill.ca/%7Elanger/558.html

Beyond the visible spectrum: infrared cameras



Beyond the visible spectrum: infrared cameras



Drawback:
Doesn’t work underwater

Beyond the visible spectrum: infrared cameras

Beyond the visible spectrum: RGBD cameras

Main ideas:

- Active sensing
- Projector emits infrared light in the scene
- Infrared sensor reads the infrared light
- Deformation of the expected pattern allows computation of the depth

Beyond the visible spectrum: RGBD cameras

Drawbacks:

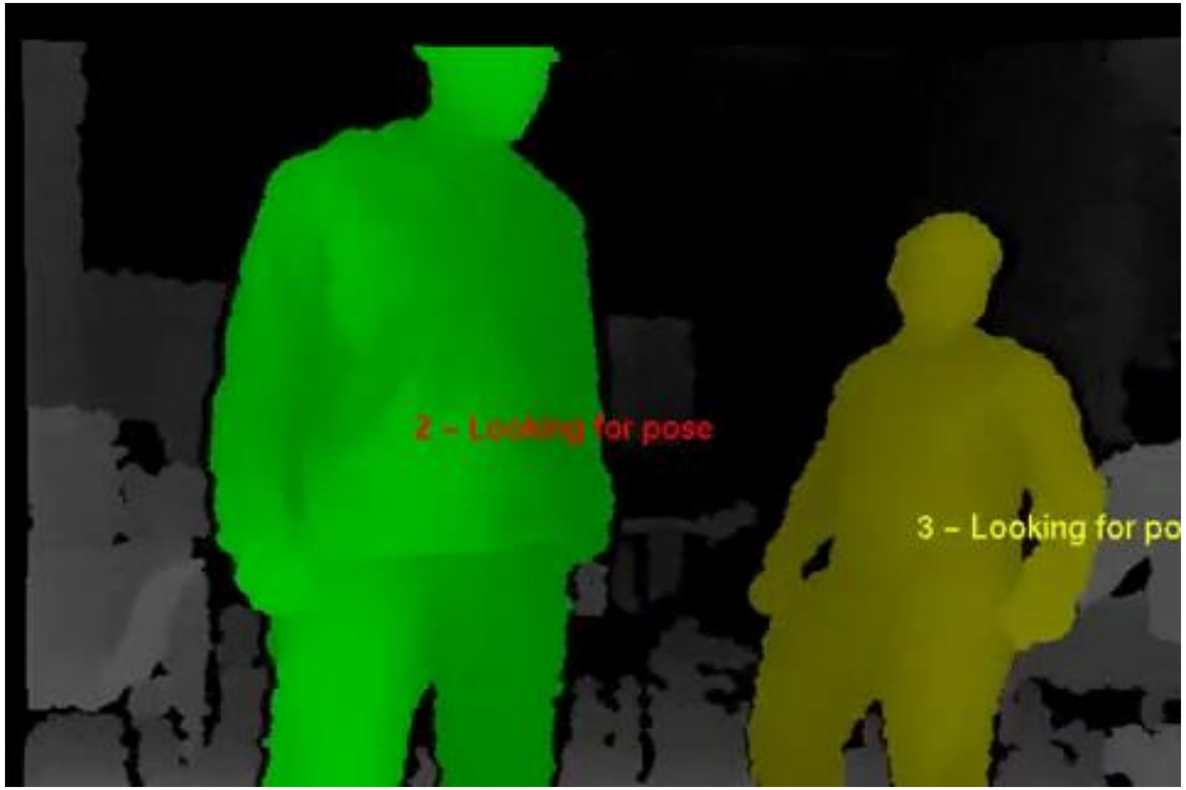
- Does not work outdoors, sunlight saturates its measurements
- Maximum range is [0.5, 8] meters

Advantages:

- Real-time depth estimation at 30Hz
- Cheap

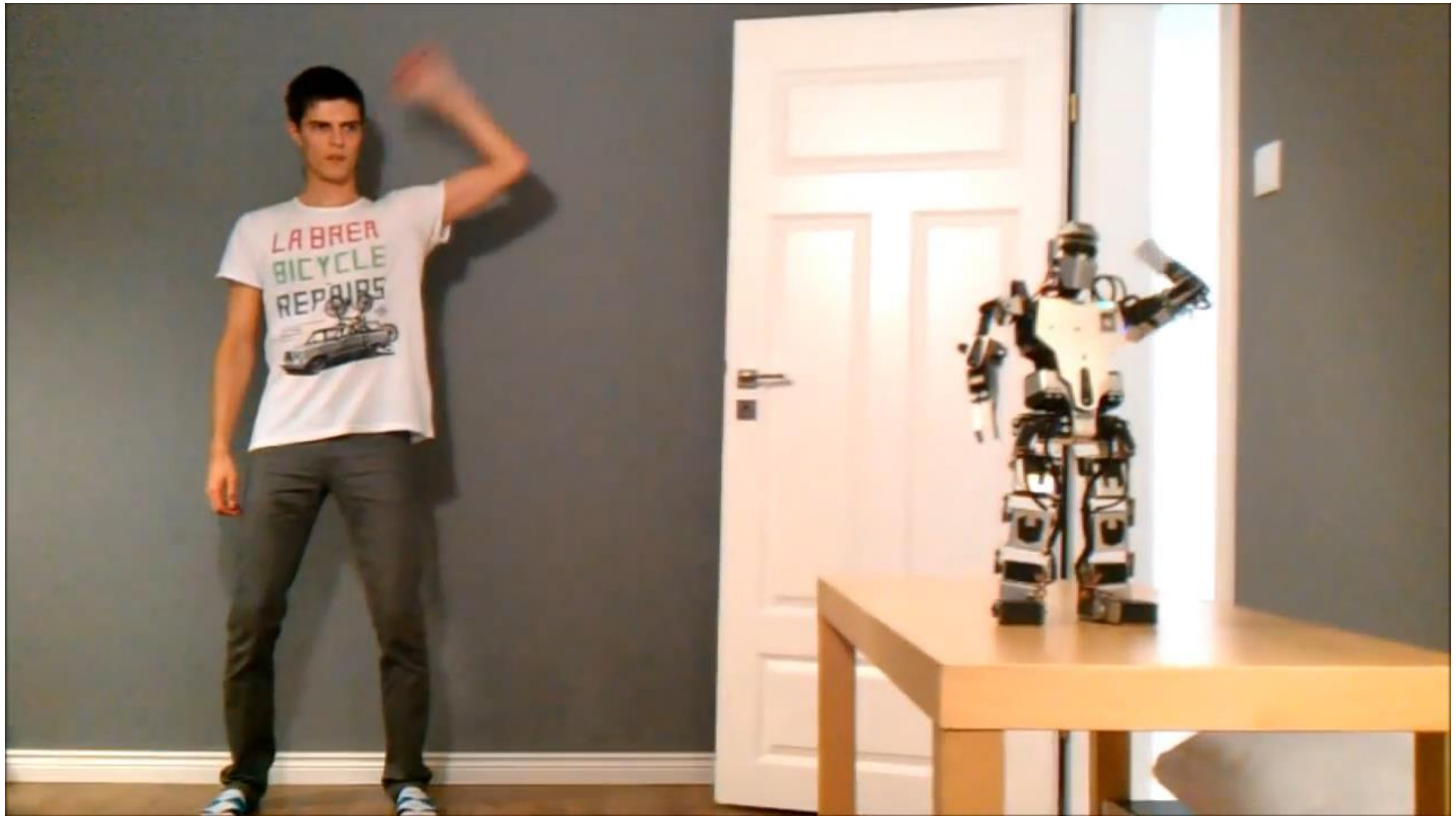
Beyond the visible spectrum: RGBD cameras

Enabled a wave of research, applications, and video games, based on real-time skeleton tracking



Beyond the visible spectrum: RGBD cameras

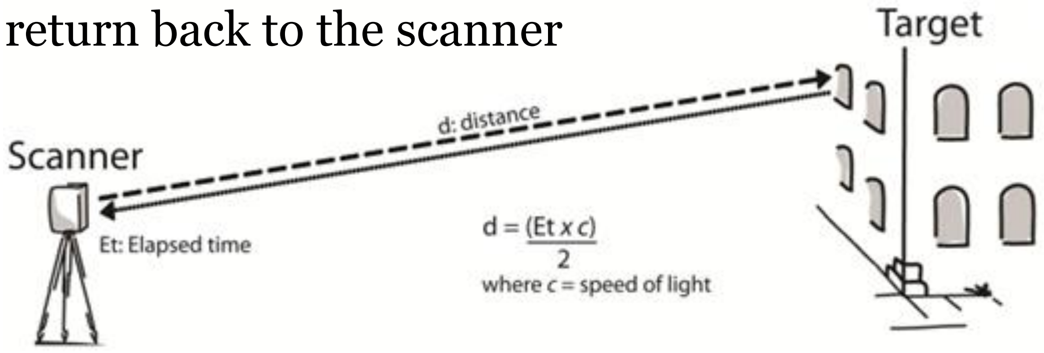
Despite their drawbacks RGBD sensors have been extensively used in robotics.



3D LIDAR (Light detection and ranging)

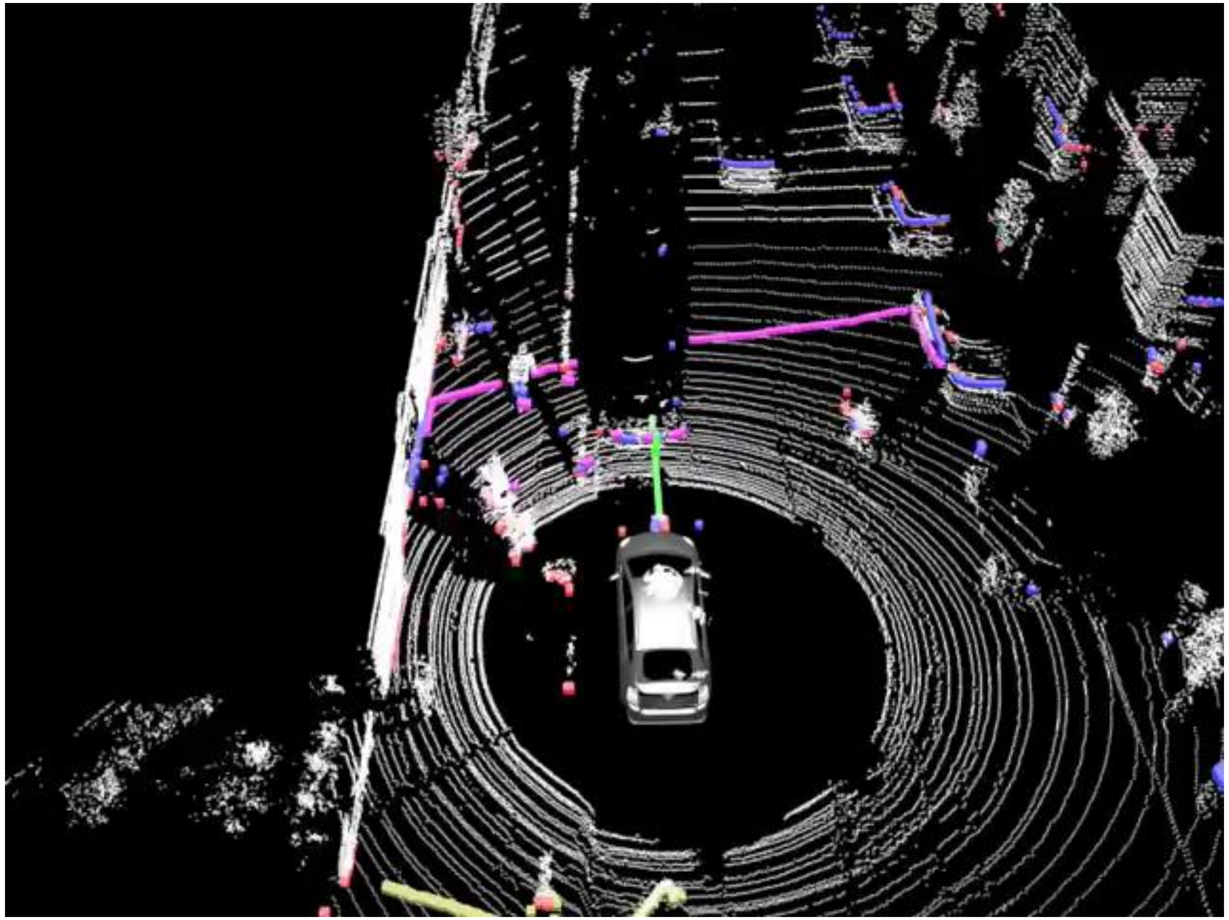
Produces a pointcloud of 3D points and intensities
- (x,y,z) in the laser's frame of reference
- Intensity is related to the material of the object that reflects the light

Works based on time-of-flight for each beam to return back to the scanner



Not very robust to adverse weather conditions: rain, snow, smoke, fog etc.

Used in most self-driving cars today for obstacle detection. Range < 100m.



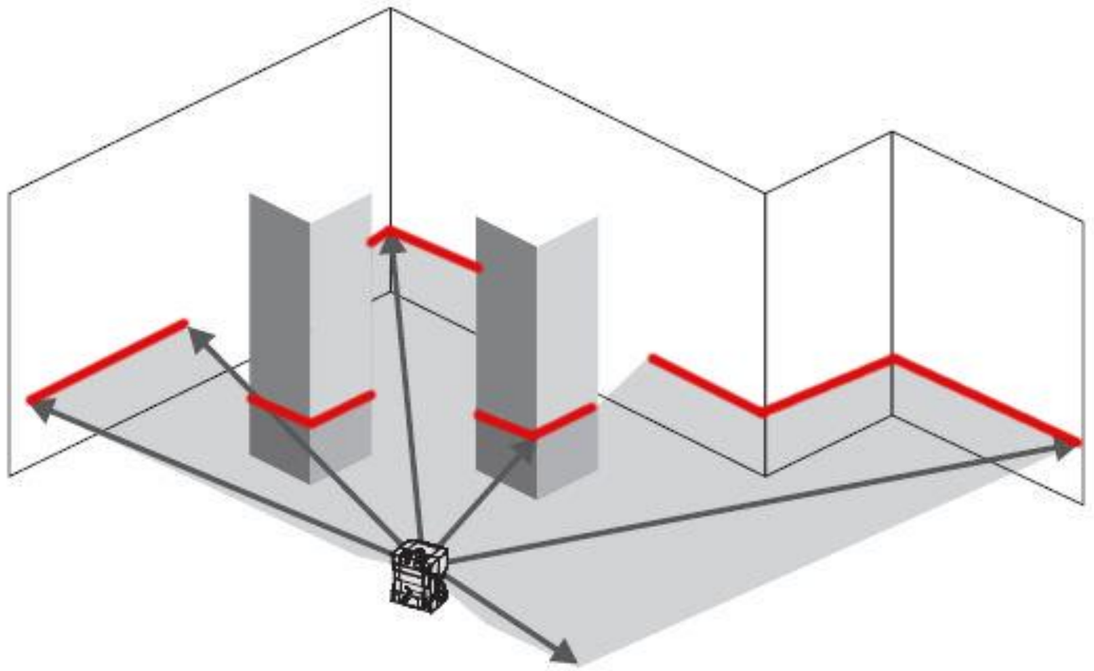
Usually around 1million points in a single pointcloud

2D LIDAR (Light detection and ranging)

Produces a scan of 2D points and intensities

- (x,y) in the laser's frame of reference
- Intensity is related to the material of the object that reflects the light

Certain surfaces are problematic for LIDAR: e.g. glass

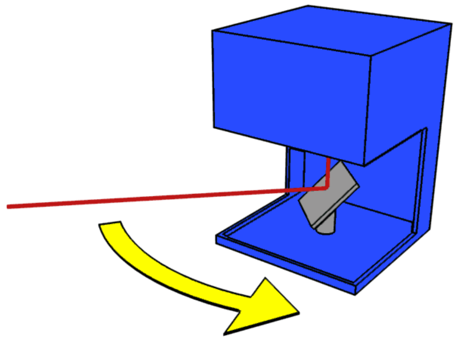


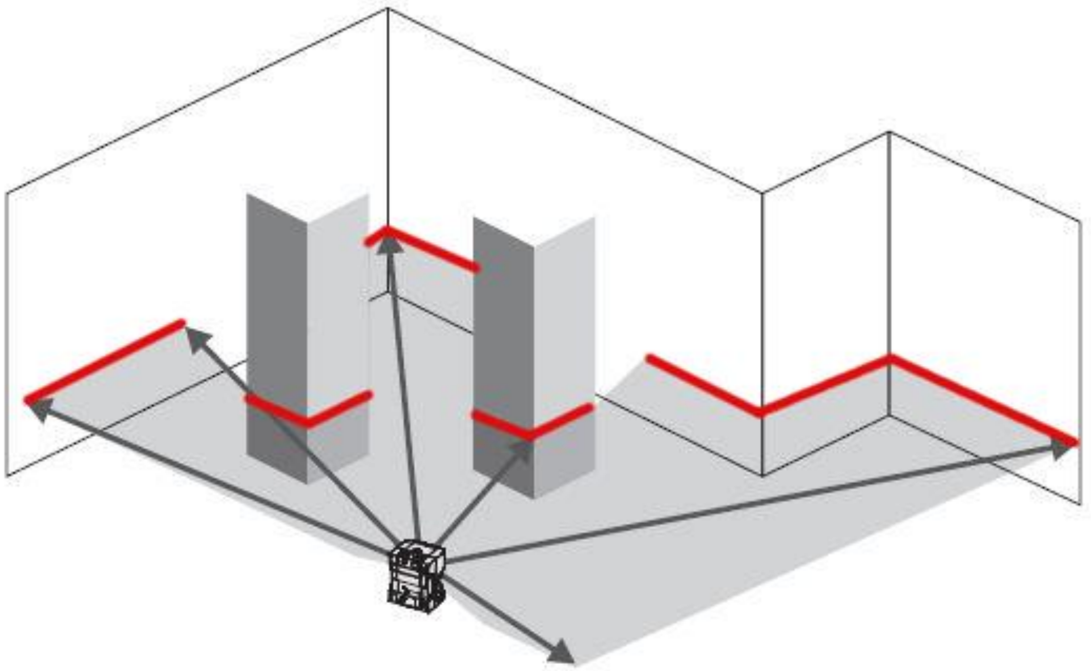
2D LIDAR (Light detection and ranging)

Produces a scan of 2D points and intensities

- (x,y) in the laser's frame of reference
- Intensity is related to the material of the object that reflects the light

Certain surfaces are problematic for LIDAR: e.g. glass





Usually around 1024 points in a single scan.

Lots of moving parts: motors quickly rotate the laser beam and once complete (angle bound reached) a scan is returned. I.e. points are not strictly speaking time-synchronized, even though we usually treat them as such.

Inertial Sensors

- Gyroscopes, Accelerometers, Magnetometers
- Inertial Measurement Unit (IMU)

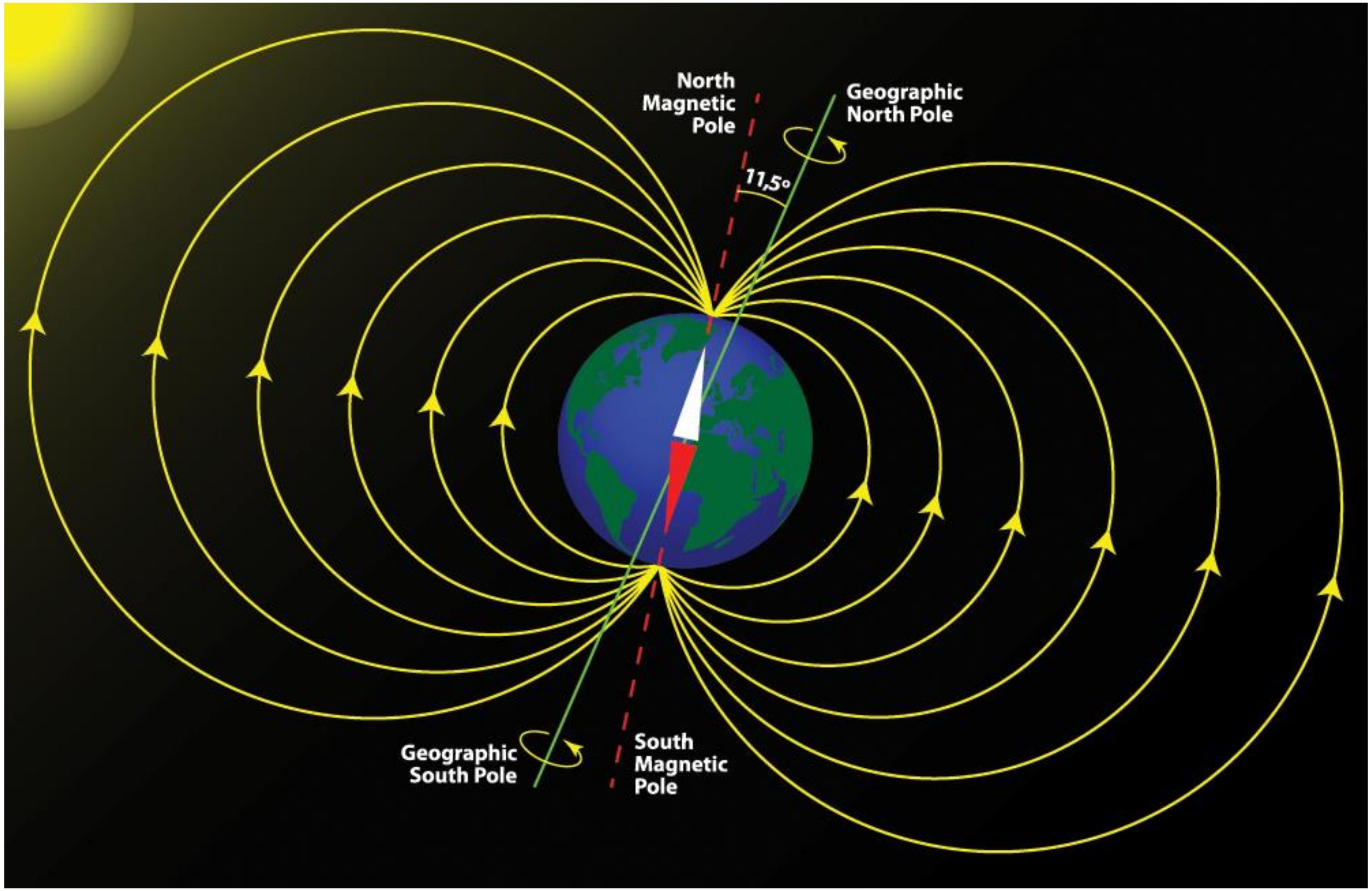
- Perhaps the most important sensor for 3D navigation, along with the GPS

- Without IMUs, plane autopilots would be much harder, if not impossible, to build

Magnetometers

Drawbacks:
- Needs careful calibration
- Needs to be placed away from moving metal parts, motors

Advantages:
- Can be used as a compass for absolute heading



Gyroscopes

- Measure angular velocity in the body frame
- Often affected by noise and bias

$$\omega_{\text{measured}}(t) = \omega_{\text{true}}(t) + b_g(t) + n_g(t)$$

- We integrate it to get 3D orientation (Euler angles, quaternions rotation matrices), but there is drift due to noise and bias

Accelerometers

- Measure linear acceleration relative to freefall (measured in g)
- A free-falling accelerometer in a vacuum would measure zero g
- An accelerometer resting on the surface of the earth would measure 1g
- Also affected by bias and noise. Usually modelled as:

$$a_{\text{measured}}(t) = R({}^{I}_{G}q(t))({}^{G}a - {}^{G}g)(t) + b_a(t) + n_a(t)$$

Where g is the gravity vector, I is the IMU body frame and G is the fixed world frame

- Double integration to get position is very noisy. Errors grow quadratically with time.

Inertial Measurement Unit

- Combines measurements from accelerometer, gyroscope, and magnetometer to output an estimate of orientation
  with reduced drift.

- Does not typically provide a position estimate, due to double integration.

- Runs at 100-1000Hz

- Expect yaw drift of 5-10 deg/hour on most modern low-end IMUs

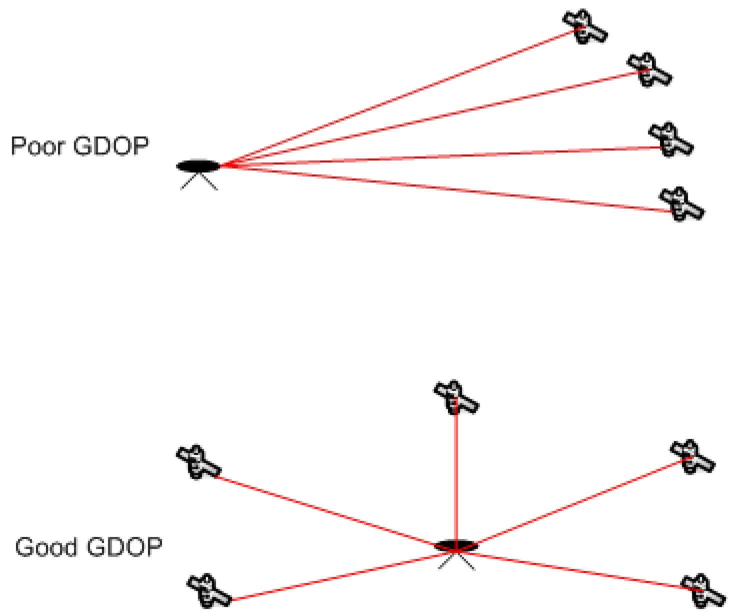
Global Positioning System: Satellites

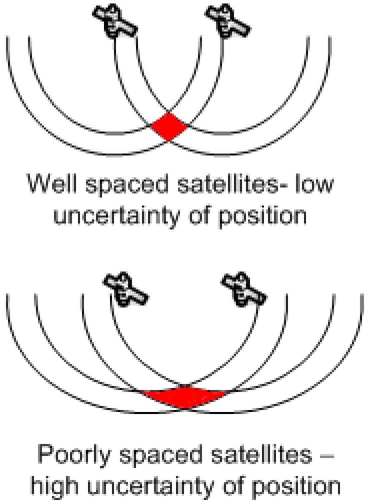
- Each GPS satellite periodically transmits:
  - [Coarse/Acquisition code] A 1023-bit pseudorandom binary sequence (PRN code), which repeats every 1 ms, unique for each satellite (no correlation with other satellites).
  - [Navigation frame] A 1500-bit packet that contains
    - GPS date, time, satellite health
    - Detailed orbital data for the satellite, accurate for the next ~4hrs
    - PRN codes and status of all satellites in the network
    - Takes 12.5mins to transmit
  - [Precision code] A 6.2-terabit code for military use.
- Carrier frequencies are 1575.42 MHz (L1) and 1227.60 MHz (L2)

Global Positioning System: Receivers

- Each (civilian) GPS receiver:
  - Knows the PRN codes for each satellite in advance
  - Correlates received PRN signal with database PRN signal → time shift → noisy distance to satellite
  - If 4 or more satellite PRN codes are received, it does **trilateration** to compute latitude and longitude

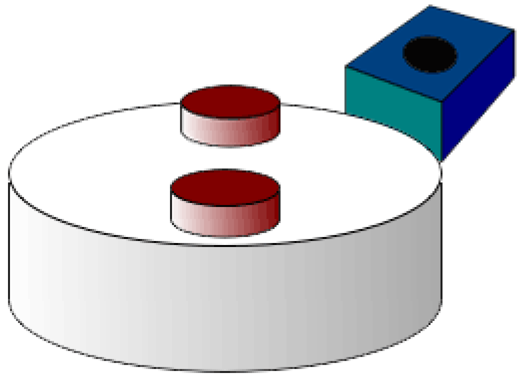
Global Positioning System: Receivers and Dilution of Precision





Geometry in 2-D (GPS Basics, 2000)

Hall Effect Sensor

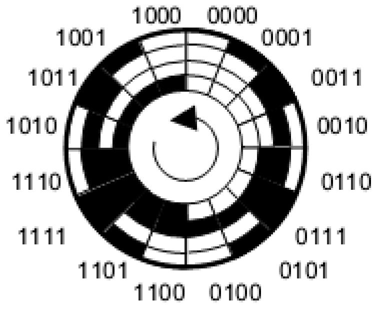


- Varies its voltage in response to a magnetic field
- Used as a proximity switch, to measure a full rotation of a wheel for example
- Used to measure rate of rotation of wheels

Rotary Encoder

- Contains an analog to digital converter for encoding the angle of a shaft/motor/axle

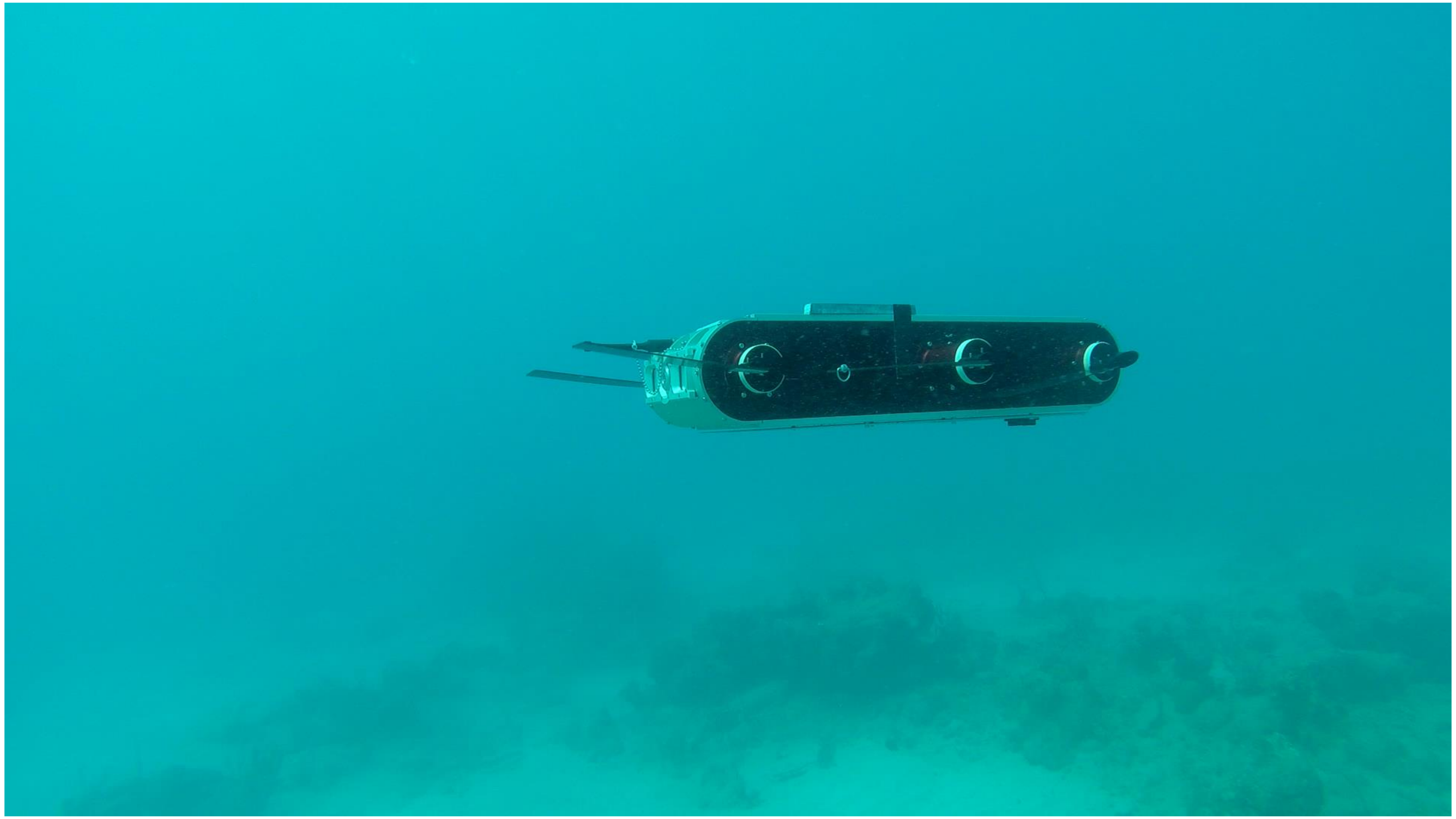
- Usually outputs the discretized absolute angle of the shaft/motor/axle



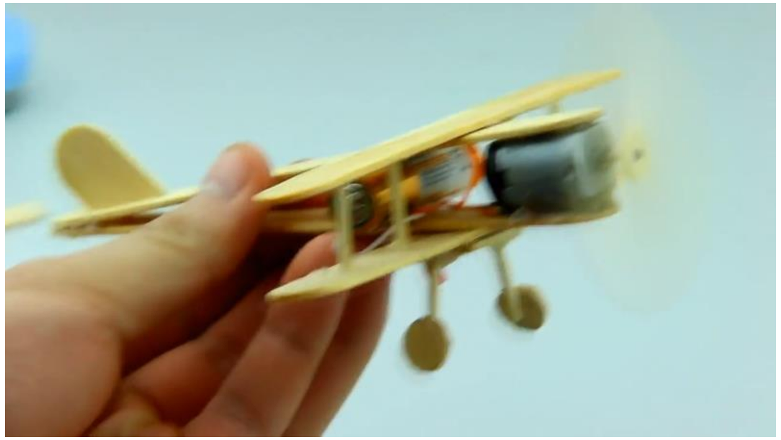
Gray code

- Useful in order to know where different shafts are relative to each other.

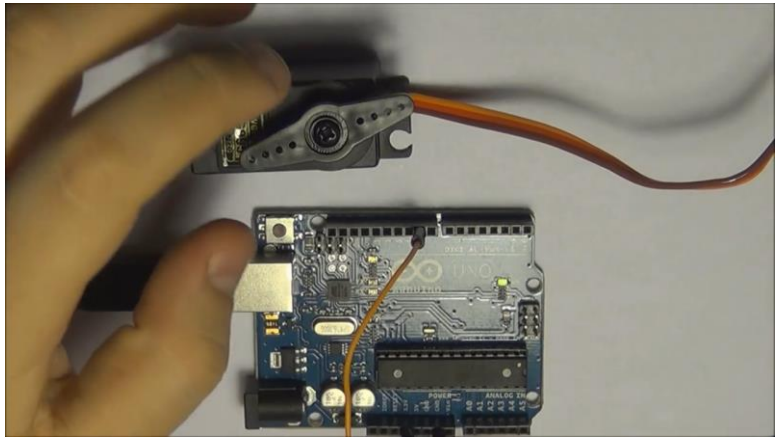
Example: flippers on the Aqua robot



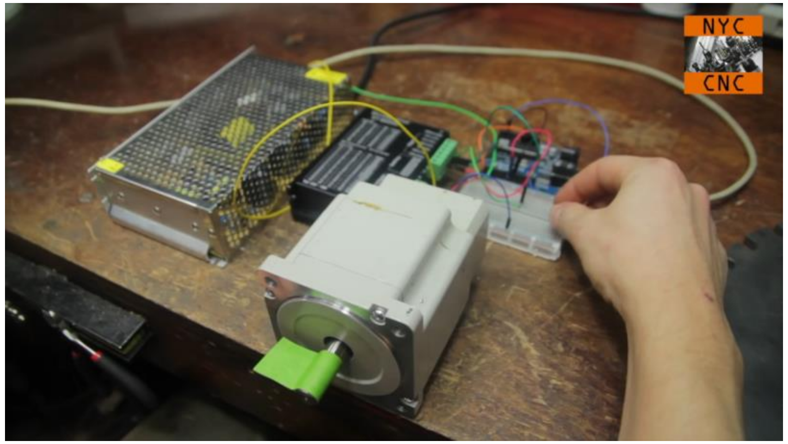
Actuators



DC (direct current) motor
They turn continuously at high RPM (revolutions per minute) when voltage is applied. Used in quadrotors and planes, model cars etc.

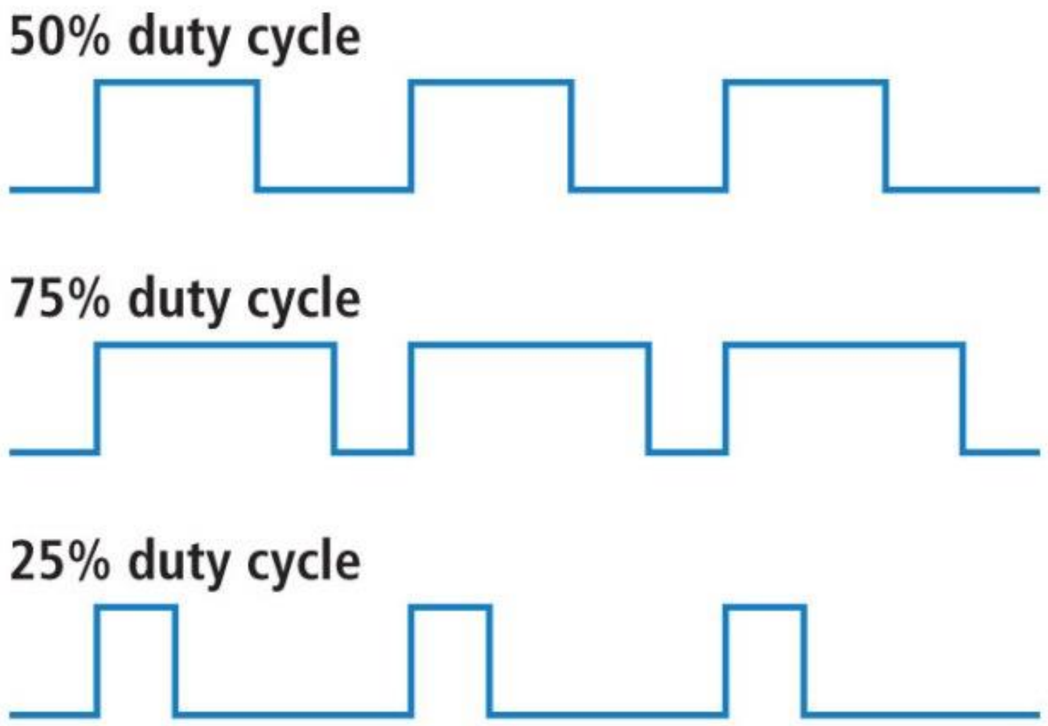


Servo motor
Usually includes: DC motor, gears, control circuit, position feedback
Precise control without free rotation (e.g. robot arms, boat rudders)
Limited turning range: 180 degrees



Stepper motor
Positioning feedback and no positioning errors.
Rotates by a predefined step angle.
Requires external control circuit.
Precise control without free rotation.
Constant holding torque without powering the motor (good for robot arms or weight-carrying systems).

Pulse Width Modulation



Used for creating analog/continuous behavior when voltage applied is discrete.
Main idea: turn on and off the motor fast enough so average voltage is the desired target.
Used in dimming LEDs, controlling the speed of DC motors, controlling the position of servo motors.