CS 3630!

Lecture 25: Drone Actions

Sensing for Quadrotor drones

1. Gyroscopes
2. Accelerometers
3. Magnetometers
4. AHRS
5. INS
6. Cameras
7. Intrinsics
8. Extrinsics
9. Projecting Points
10. Example: Stereo

Gyroscopes

From wikipedia

- Used to be mechanical: spinning wheel
- Now MEMS
- Measures angular velocity
- Need to integrate for attitude

$$R_b^n(t) = R_b^n(0) \int_{\tau=0}^{t} \exp \hat{\omega}(\tau) d\tau$$

- Challenge: noise and bias

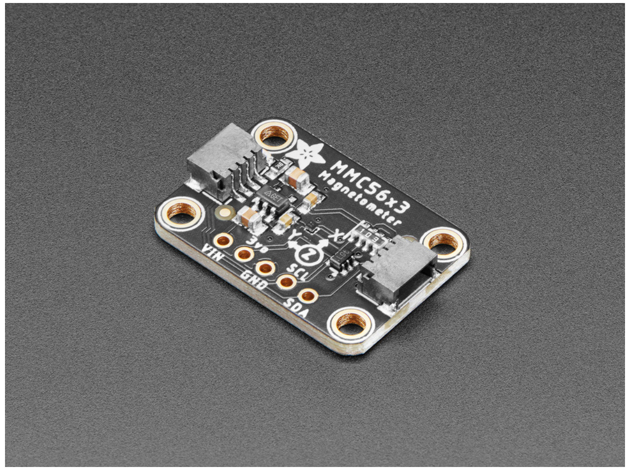
Accelerometers

From [wikipedia]

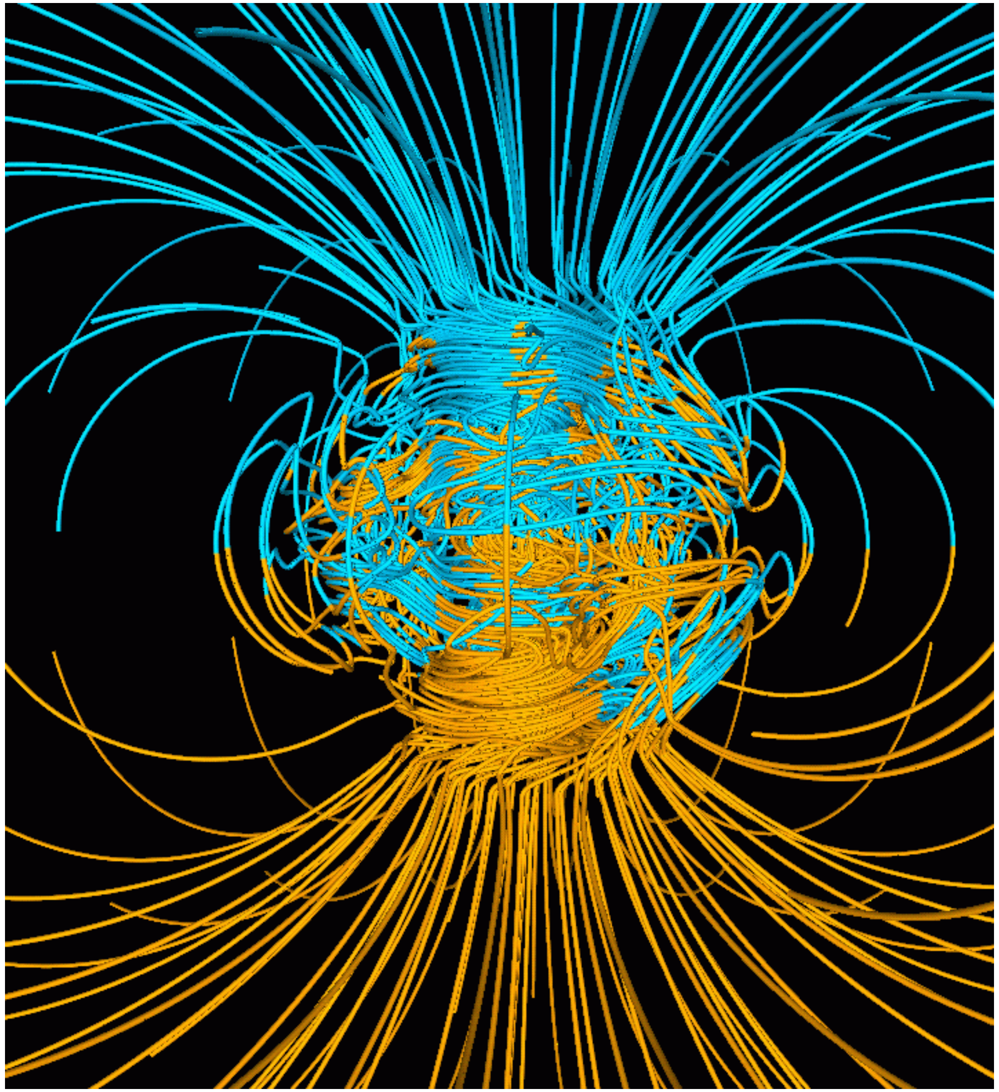
- Measures force, translated to acceleration
- Double integration: very challenging!
- In phones: more useful to “aid” gyroscope.

Magnetometers

- Earth magnetic field is 3D and complex
- Unreliable near metal
- Still helpful



From Adafruit ($5.95)



Public Domain,
https://commons.wikimedia.org/w/index.php?curid=1712490

AHRS

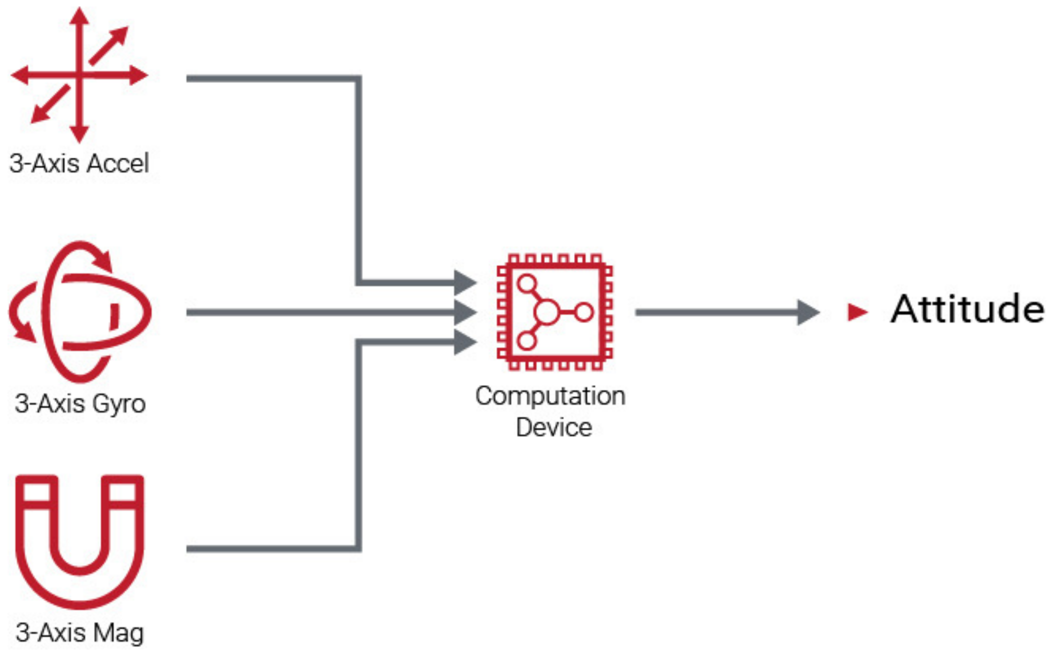
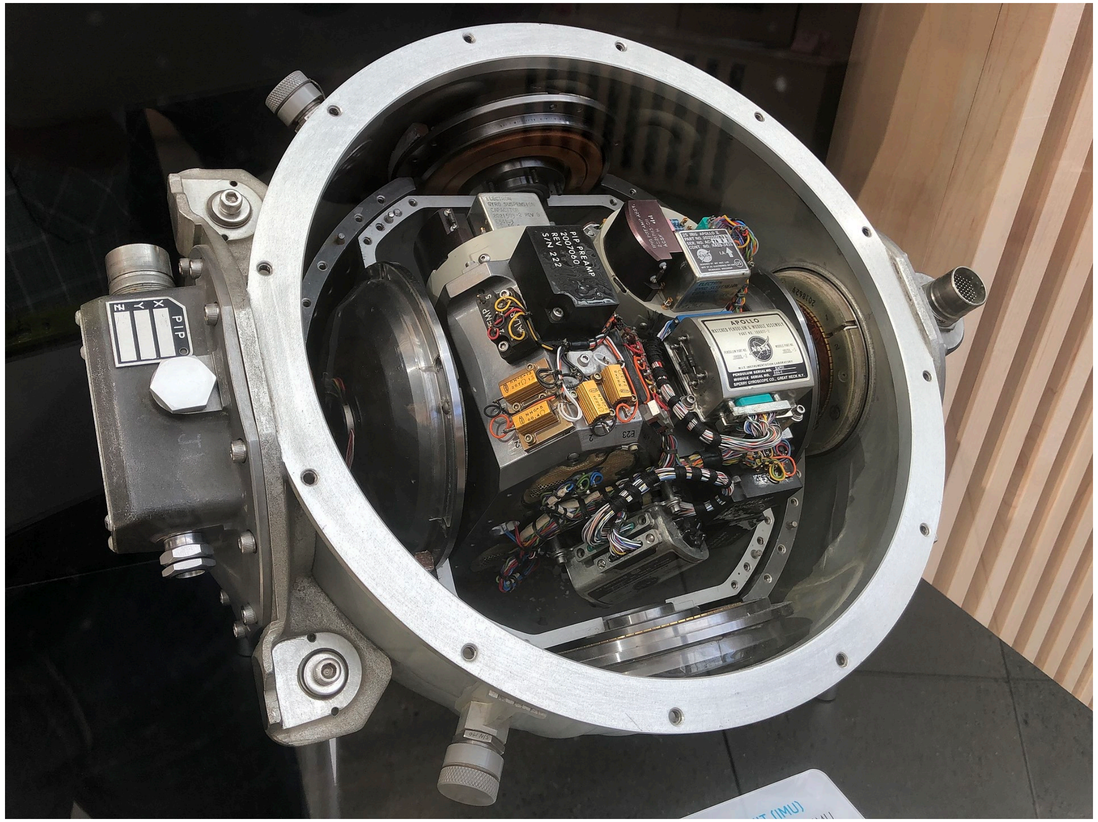


Image from VectorNav course

- AHRS = Attitude and heading reference system
- Gyro = accurate and fast, but attitude drifts!
- Accelerometer points to gravity = 2 out of 3DOF
- Magnetometer provides (complicated) signal on heading

INS



Apollo INS By ArnoldReinhold - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=82248569

- INS = Inertial Navigation System
- Also tries to
  - Integrate accelerometer
  - estimate accelerometer biases
- Needs either:
  - Very very good IMUs (military)
  - Aiding with GPS or other correction signal, e.g., a map!
- Now: strapdown-MEMS:

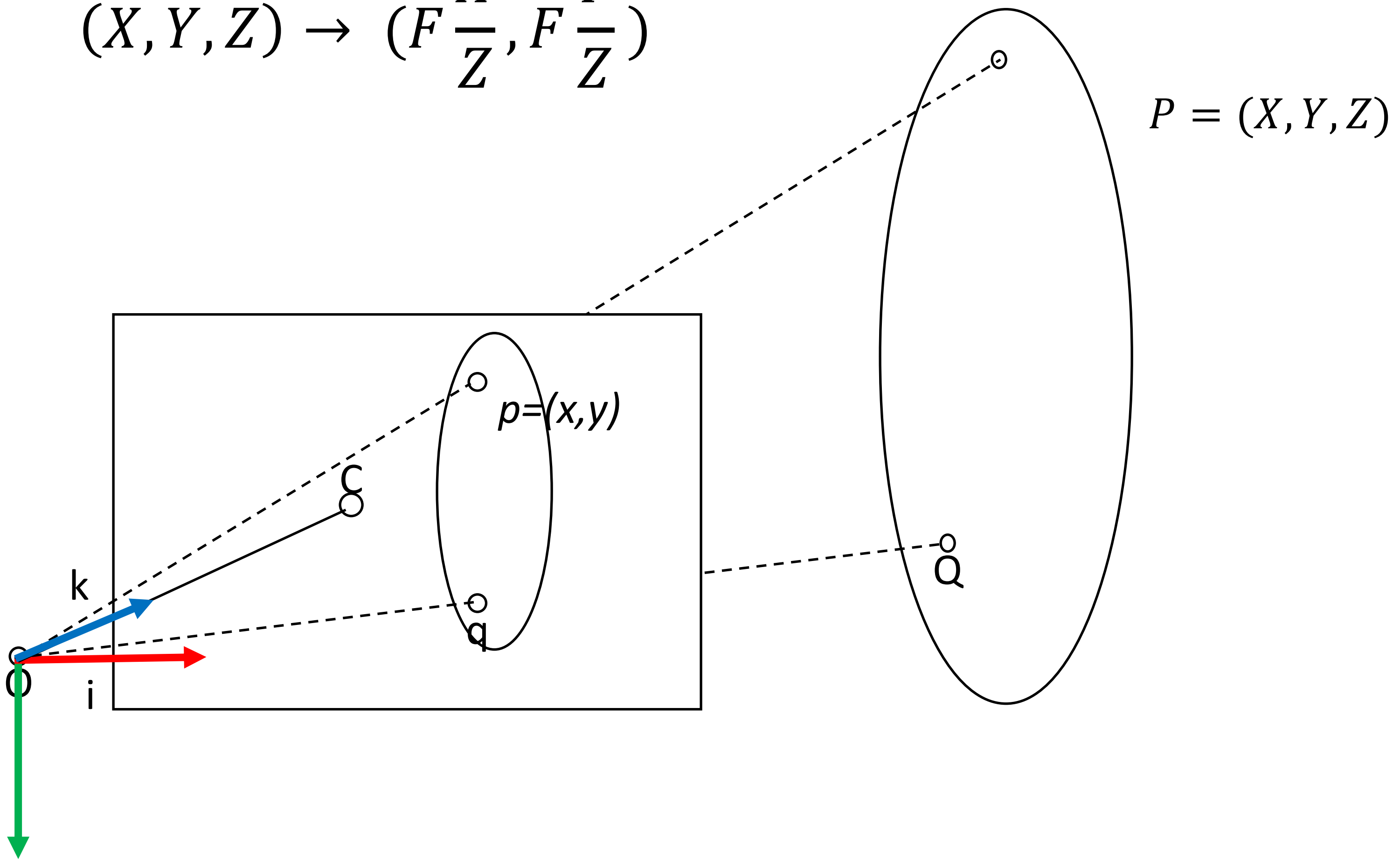
Cameras

- Light-weight & cheap!
- Passive: low power & stealth
- Supports:
  - Visual odometry
  - Localization
  - Visual SLAM
- For Skydio:
  - Tracking people
  - 3D reconstruction

Pinhole model

Review: pinhole equation:

$$(X, Y, Z) \rightarrow (F\frac{X}{Z}, F\frac{Y}{Z})$$



Intrinsic Camera Calibration

From image-plane coordinates to sensor coordinates

We define $(x,y)=(X/Z, Y/Z)$

To convert from image-plane coordinates to sensor coordinates u, v

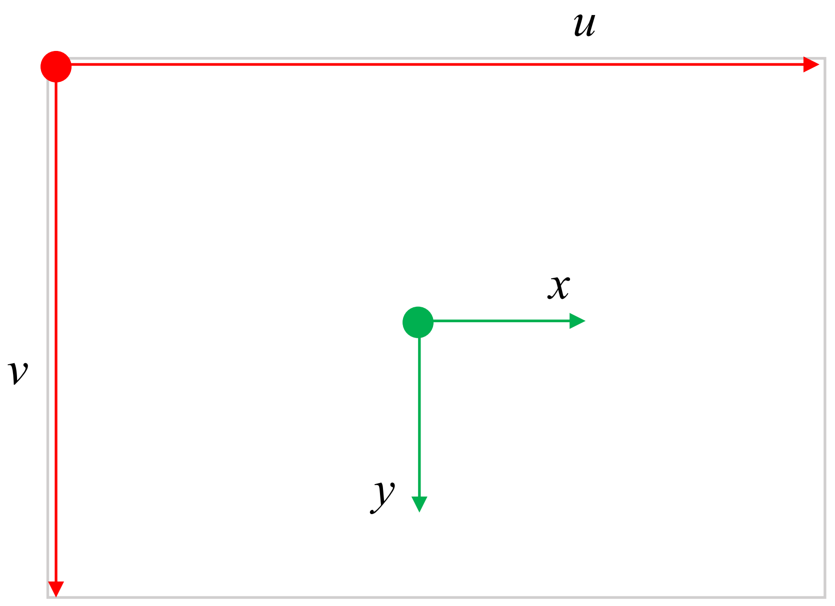
- Scale x by focal length and pixel width
- Scale y by focal length and pixel height
- Shift coordinates by u_0, v_0:

$$u = u_0 + \alpha F \frac{X}{Z}, \qquad v = v_0 + \beta F \frac{Y}{Z}$$

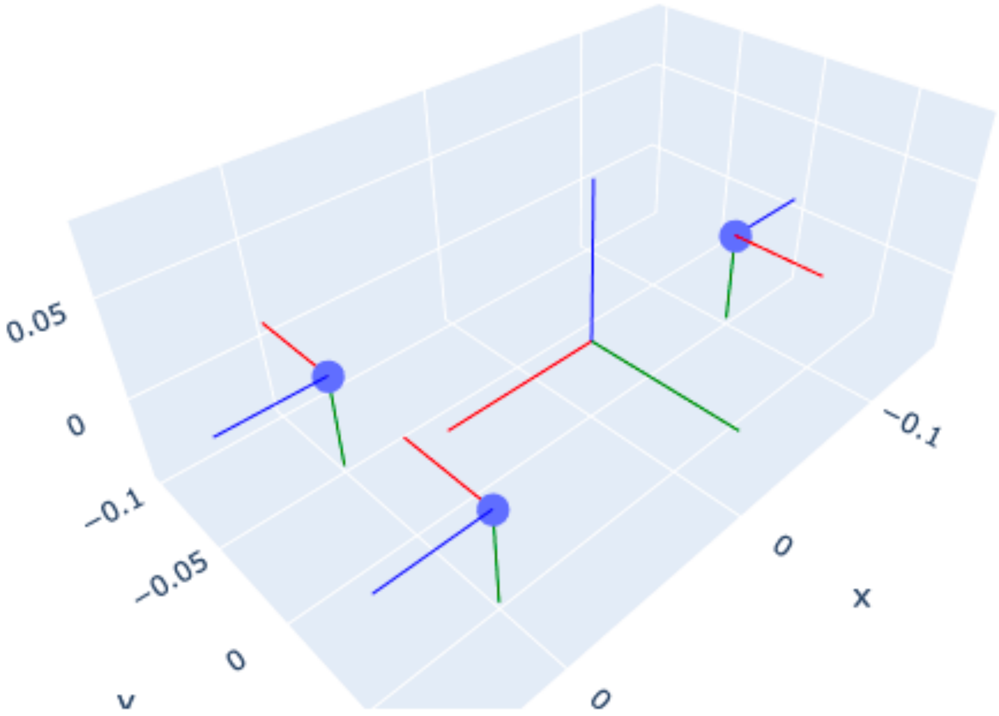
If the camera happens to have square pixels, then $\alpha = \beta$ and we can simplify this to

$$u = u_0 + f x, \qquad v = v_0 + f y$$

Camera calibration is used to determine the values of u_0, v_0 and f.



Extrinsic Calibration



- A drone might have multiple cameras, e.g., 3 in this example
- Calibrate extrinsics for each:
  - Position t_c^b in body frame
  - Orientation R_c^b in body frame
- Always: think about columns!

```
F,L,U = np.eye(3)
bTc1 = gtsam.Pose3(gtsam.Rot3(-L,-U,F), t1)
bTc2 = gtsam.Pose3(gtsam.Rot3(-L,-U,F), t2)
bTc3 = gtsam.Pose3(gtsam.Rot3(L,-U,-F), t3)
```

Projecting Points

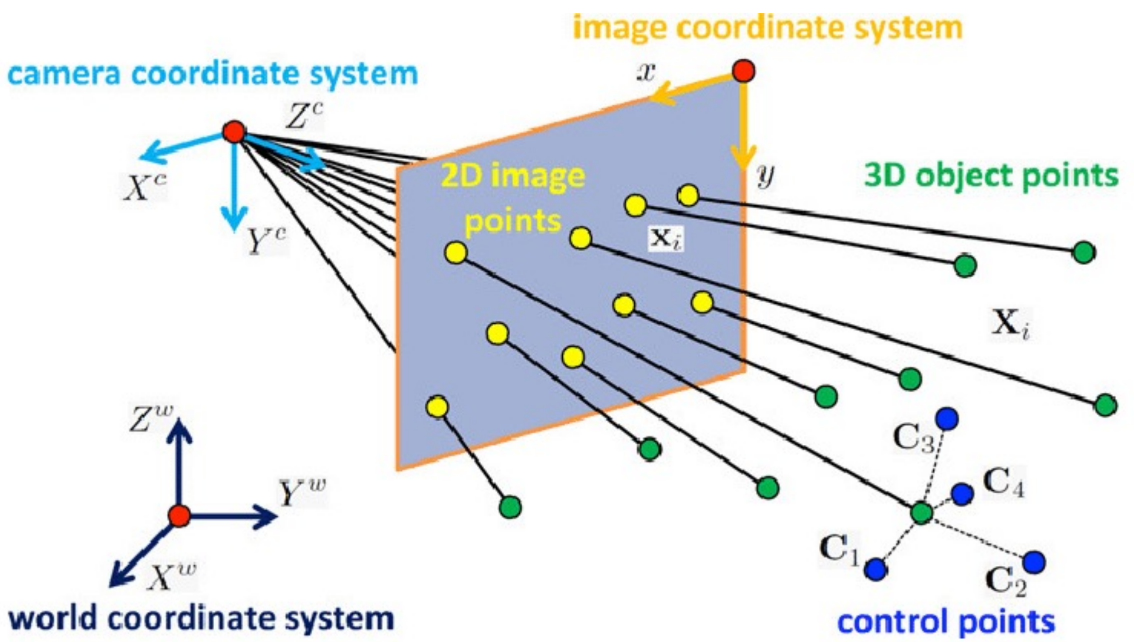


Image by Boris Jutzi et al

- We know how to go from 3D camera coordinates to pixels:

$$u = u_0 + f\frac{X^c}{Z^c} \quad v = v_0 + f\frac{Y^c}{Z^c}.$$

- So, before projecting: 2 steps:
  - convert from navigation to body frame: $P^b = (R_b^n)^T(P^n - t_b^n)$
  - convert from body to camera frame: $P^c = (R_c^b)^T(P^b - t_c^b)$

Worked Example: Stereo

```
E,N,U = np.eye(3)
ntb = gtsam.Point3(100, 300, 10)
nRb = gtsam.Rot3(N,-E,U) # flying north, left of drone facing west
nTb = gtsam.Pose3(nRb, ntb)
```

- Specify body in ENU nav frame

```
wP = gtsam.Point3(103,310,12)
bP = nTb.transformTo(wP)
print(f"bP = {bP} in (F,L,U) body frame")
c1P = bTc1.transformTo(bP)
print(f"c1P = {c1P} in camera frame 1")
c2P = bTc2.transformTo(bP)
print(f"c2P = {c2P} in camera frame 2")
```

- Convert point in nav to body
- Convert body to c1, c2

```
bP = [10. -3.  2.] in (F,L,U) body frame
c1P = [ 3.05 -1.99  9.9 ] in camera frame 1
c2P = [ 2.95 -1.99  9.9 ] in camera frame 2
```

- Apply intrinsics

```
w, h, f = 640, 480, 300
u0, v0 = float(w/2), float(h/2)
u1, v1 = u0 + f * c1P[0]/c1P[2], v0 + f * c1P[1]/c1P[2]
print(f"u1,v1 = {np.round([u1,v1],1)} in image 1")
u2, v2 = u0 + f * c2P[0]/c2P[2], v0 + f * c2P[1]/c2P[2]
print(f"u2,v2 = {np.round([u2,v2],1)} in image 2")
```

```
u1,v1 = [412.4 179.7] in image 1
u2,v2 = [409.4 179.7] in image 2
```

- Check stereo result:

$$Z = B\frac{f}{d} = 0.1\frac{300}{3} = 10$$

Summary

1. Gyroscopes
2. Accelerometers
3. Magnetometers
4. AHRS
5. INS
6. Cameras
7. Intrinsics
8. Extrinsics
9. Projecting Points
10. Example: Stereo