Announcements

Final project upcoming deadlines:

- **Nov 13,** a short video of your progress.
  Submit via Compass, 10% of the final project total.

- **Dec 16,** 7-11pm in Siebel 4240.
  Final project presentations and Open House for press!

Grades are out for:

- Midterm 1
- Project abstract and picture-title

Tracking Systems in VR: Estimating 3D Orientation

Integrate sensor readings to estimate orientation:

Recall 2D: $\hat{\theta} = \theta_0 + \sum_{i=1}^{k} \Delta\hat{\theta}_i = \Delta\hat{\theta}_k + \Delta\hat{\theta}_{k-1} + \ldots + \Delta\hat{\theta}_1 + \theta_0$

$\Delta\hat{\theta}_i = \hat{\omega}_i \Delta t$

Now 3D: $\hat{Q} = \Delta\hat{Q}_k \circ \Delta\hat{Q}_{k-1} \circ \ldots \circ \Delta\hat{Q}_2 \circ \Delta\hat{Q}_1 \circ Q_0$

Each $\Delta\hat{Q}_i = \mathrm{Quat}(\hat{\vec{v}}_i, \Delta\hat{\theta}_i)$

$\hat{\vec{v}}_i = \frac{\hat{\vec{\omega}}_i}{\|\hat{\vec{\omega}}_i\|}$, $\Delta\hat{\theta}_i = \|\hat{\vec{\omega}}_i\| \Delta t$

Problem: Drift or dead reckoning

Estimating 3D Orientation: Drift Correction

Separate the rotational drift error into **two components:**

1) Tilt error: Pitch + roll

To correct: need a gravity or "up" vector

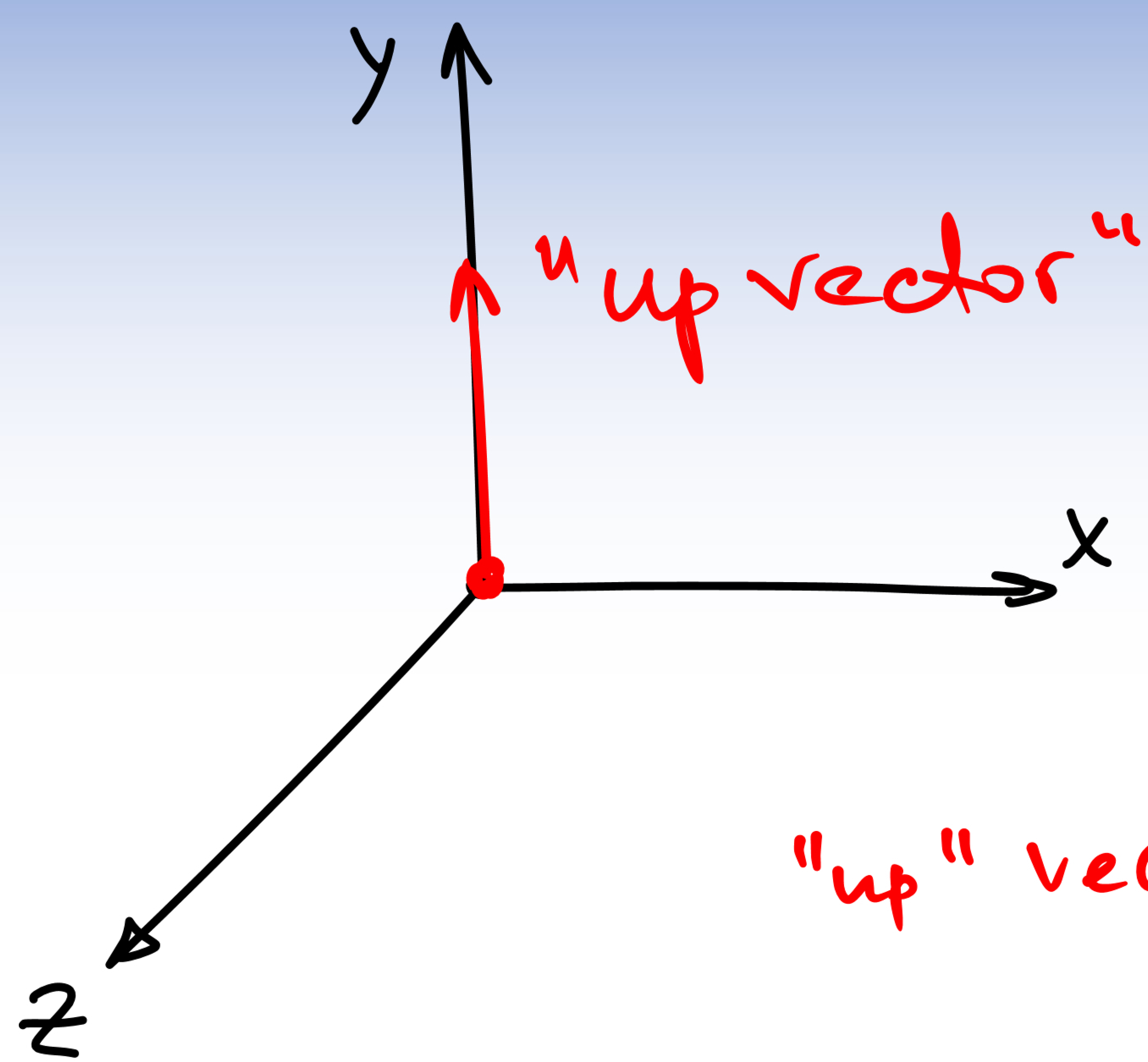
2) Yaw error

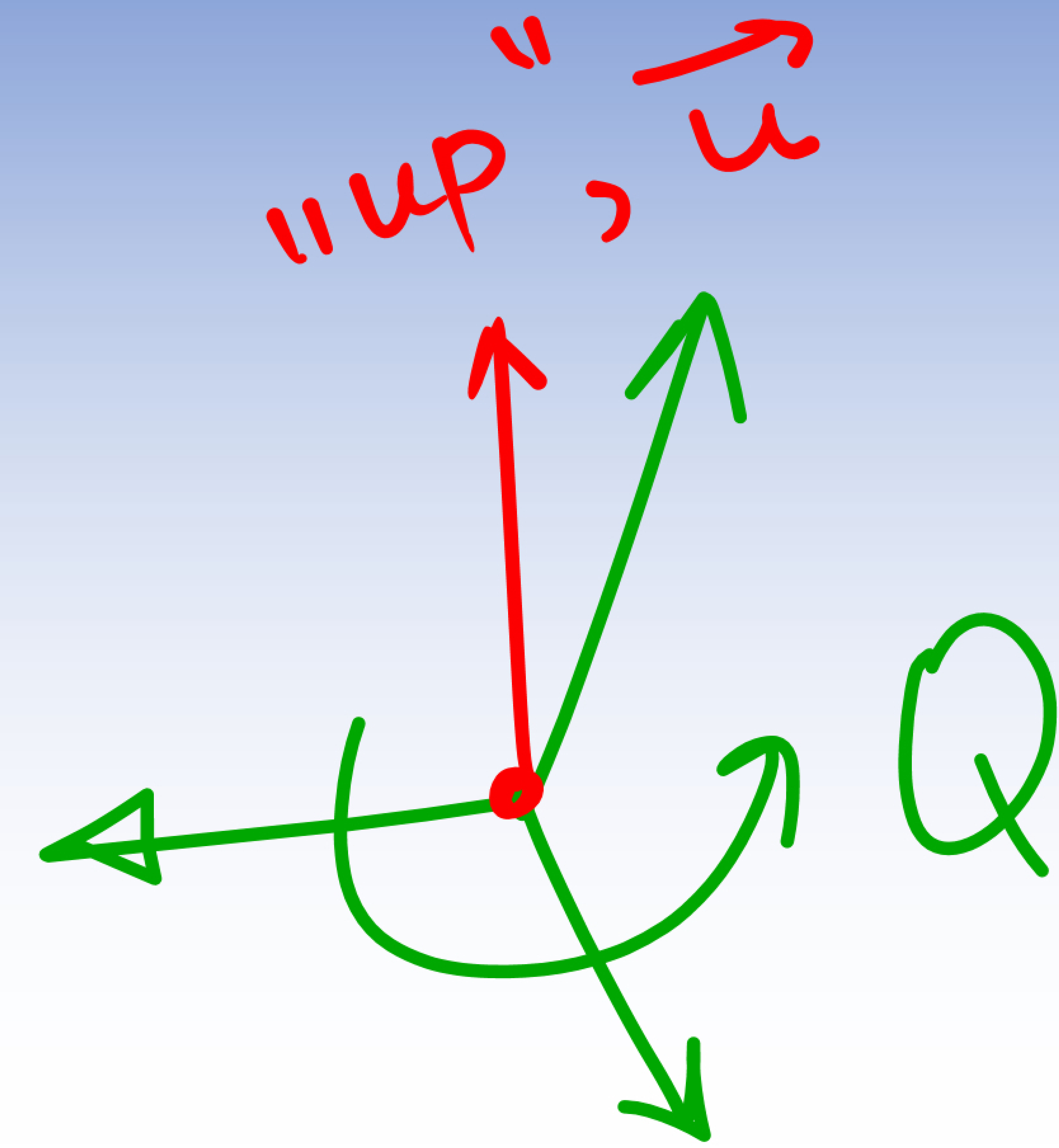
To correct: need a "compass"

- Complementary Filters on SO(3), Mahoney, 2008

- Head Tracking for the Oculus Rift, ICRA 2014
  S. LaValle, A. Yershova, M. Katsev, and M. Antonov

Use ~~Accelerometer~~ perfect "up" sensor to Correct "Tilt Error"



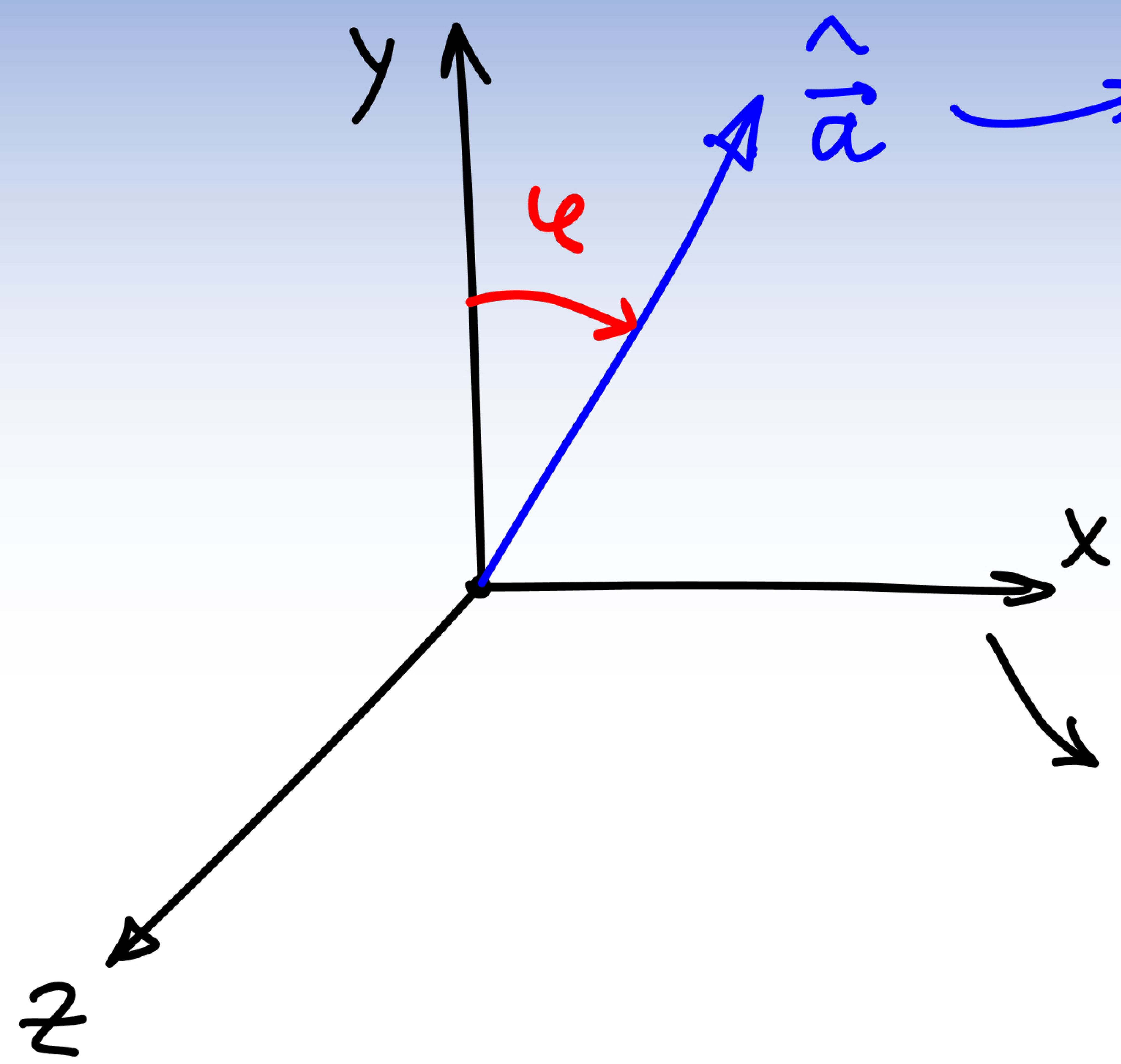


"up" vector is always

What is $Q\vec{u}$? Assume Q is accurate.

What if $Q\vec{u}$ is not aligned with y-axis?

Use ~~Accelerometer~~ perfect "up" sensor to Correct "Tilt Error"



$\hat{\vec{a}}$ → Apply $\hat{Q}_k$ to ~~accelerometer~~ perfect "up" reading

$\hat{\vec{a}} = (\hat{a}_x, \hat{a}_y, \hat{a}_z)$

→ Estimated world frame

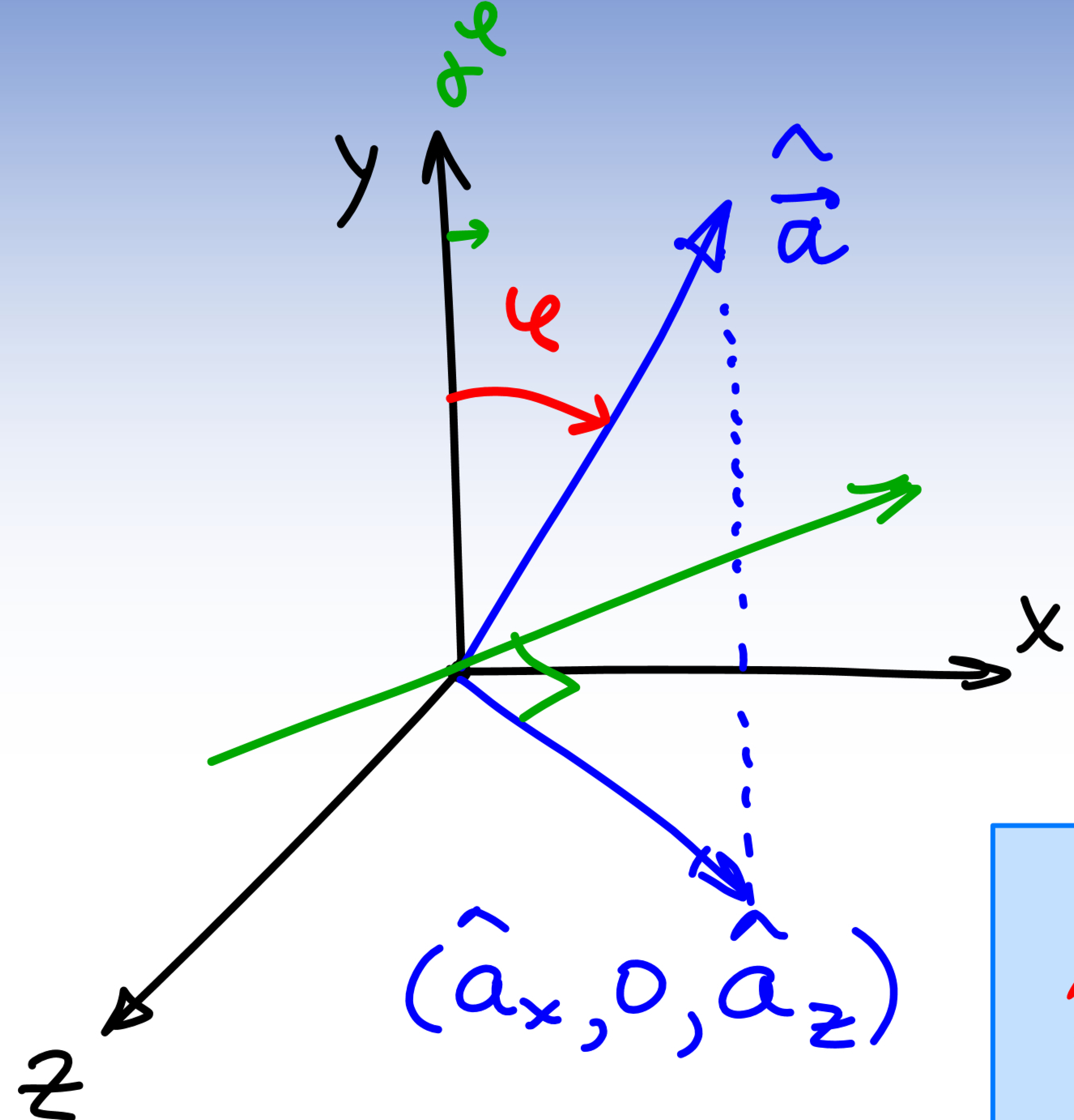
Tilt axis in XZ plane:

$\vec{V}_{tilt} = (\qquad\qquad)$

Now rotate by φ about $\vec{V}_{tilt}$ to fix $\hat{Q}_k$

$\hat{Q}_{corrected} = \qquad\qquad \circ \hat{Q}_k$

Use ~~Accelerometer~~ perfect "up" sensor to Correct "Tilt Error"



Complementary filter:

In each Δt,

rotate by

about $\vec{V}_{tilt}$ to fix $\hat{Q}_k$

recompute every Δt

$$\hat{Q}_{corrected} = Quat(\vec{v}_{tilt}, \varphi) \circ \hat{Q}_k$$

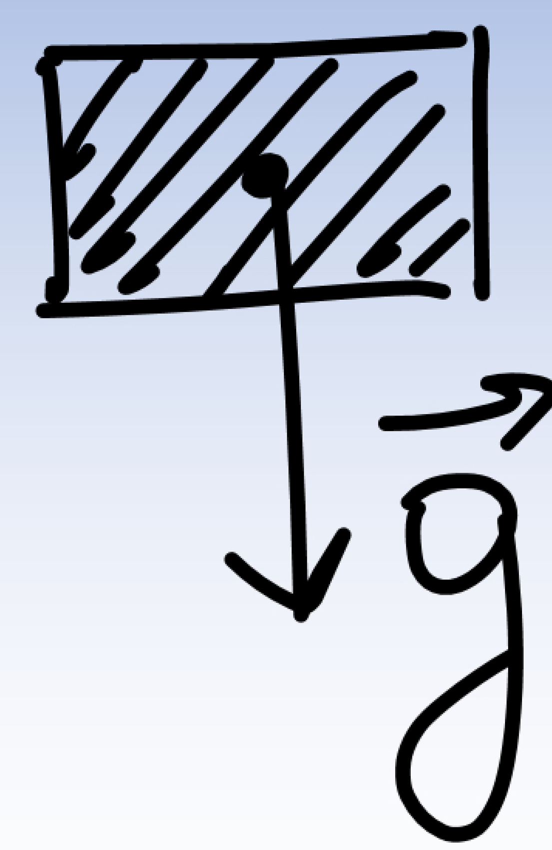
Gain coefficient $\alpha > 0$, $\alpha \approx 0$ (ex. $\alpha = 0.0001$)

α needs to be large enough to

but small enough to

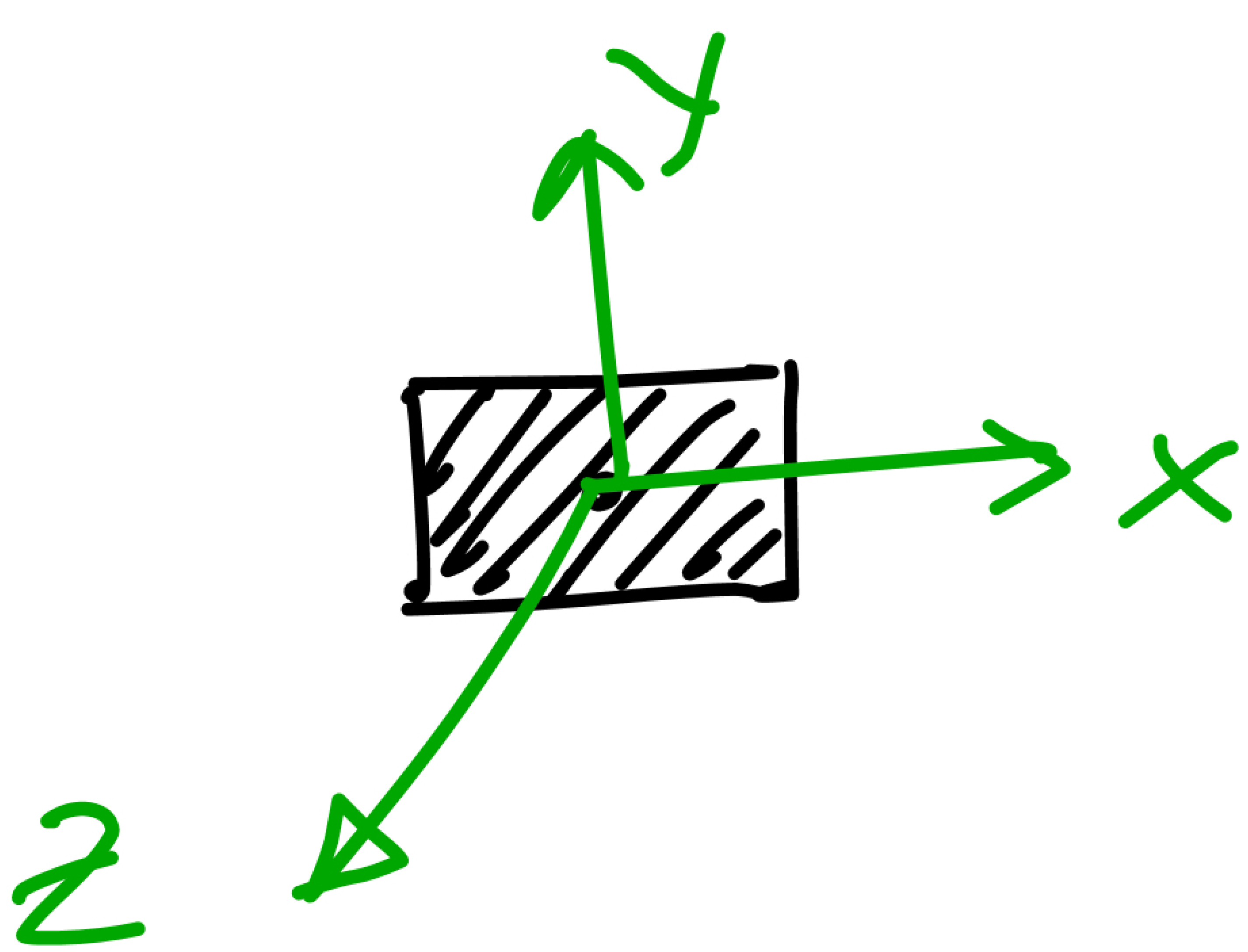
What Does Accelerometer Measure?

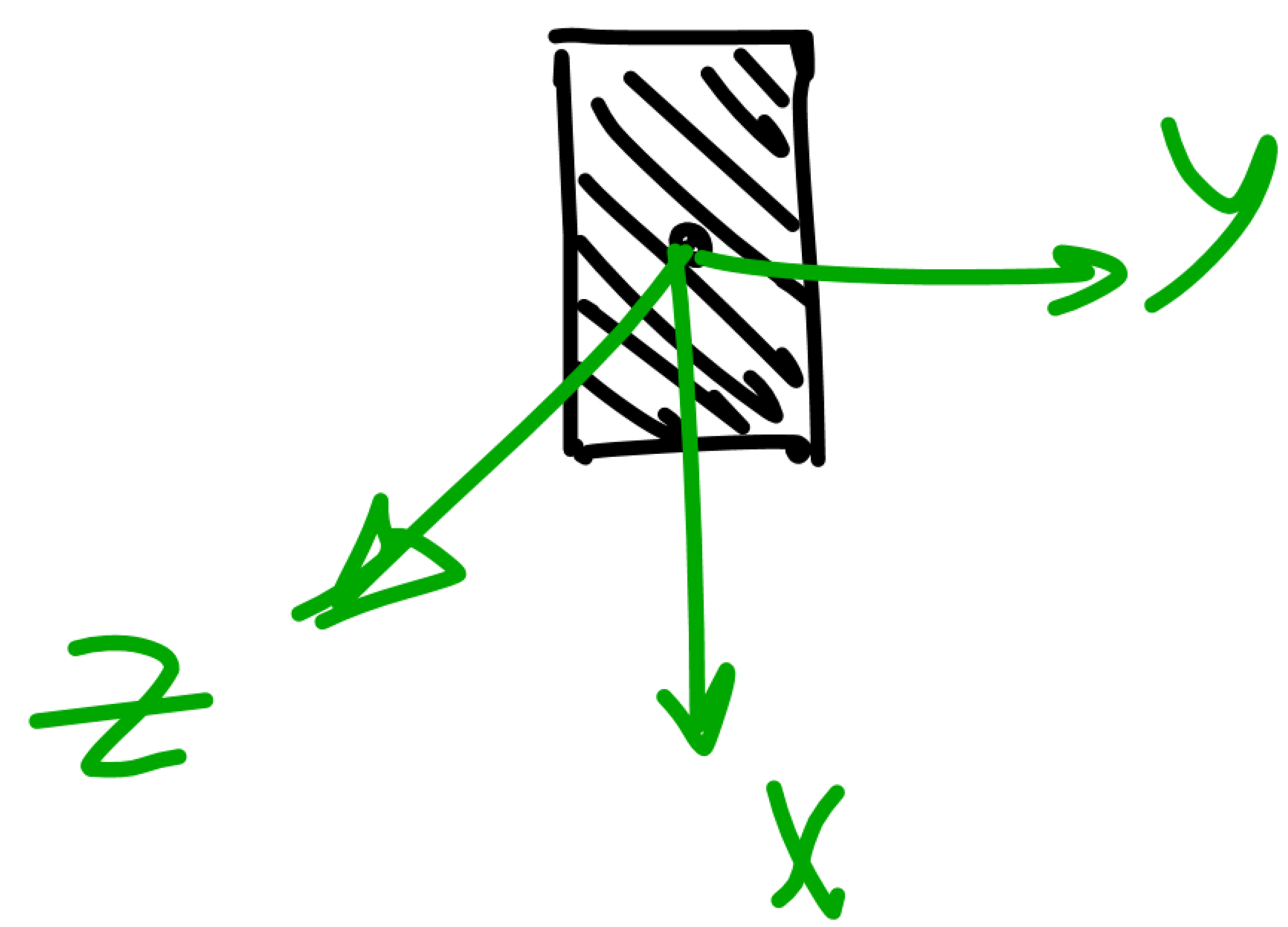
Rift is free falling:

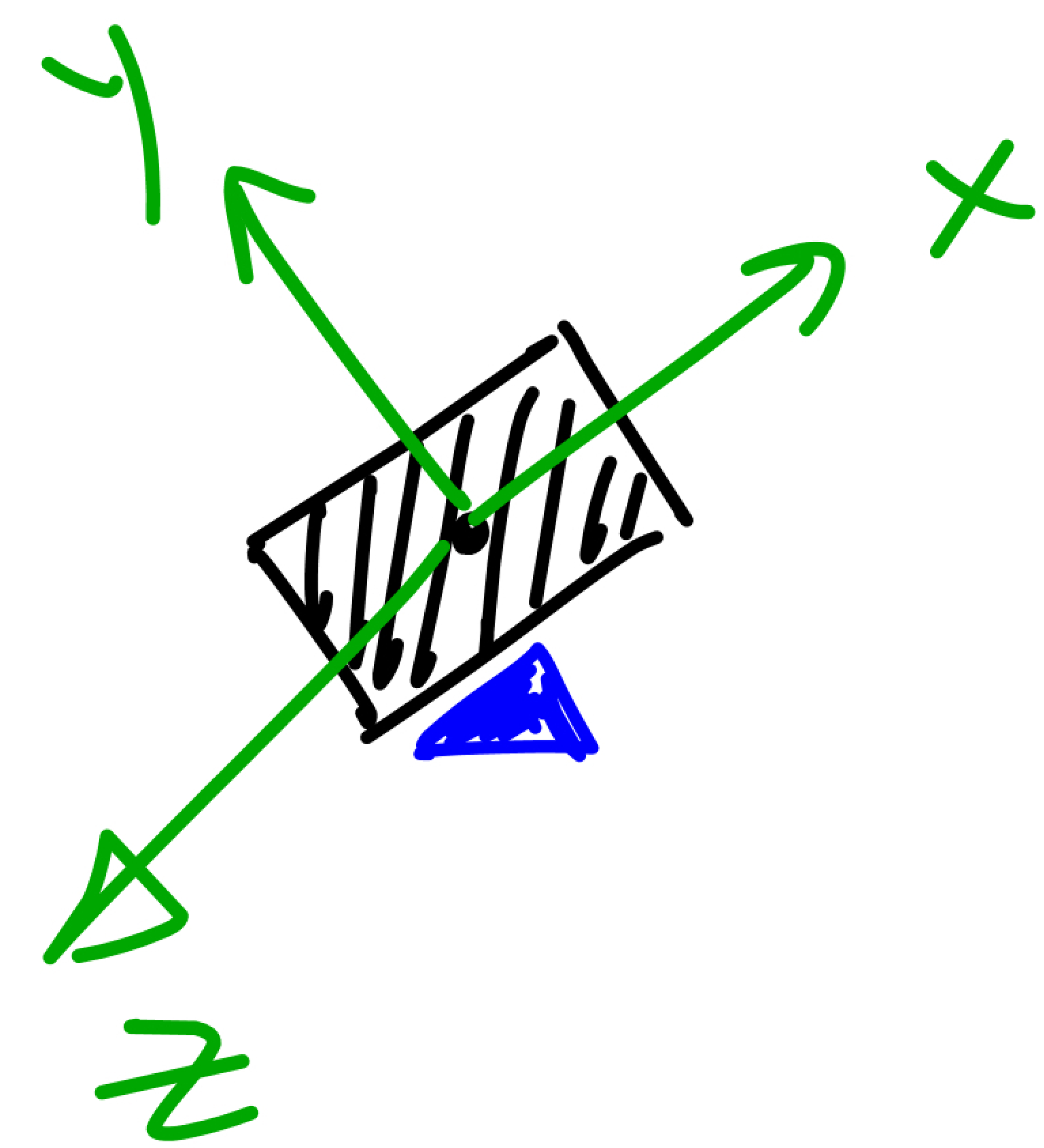


$\vec{a} = \quad)$

Rift lying on a side:

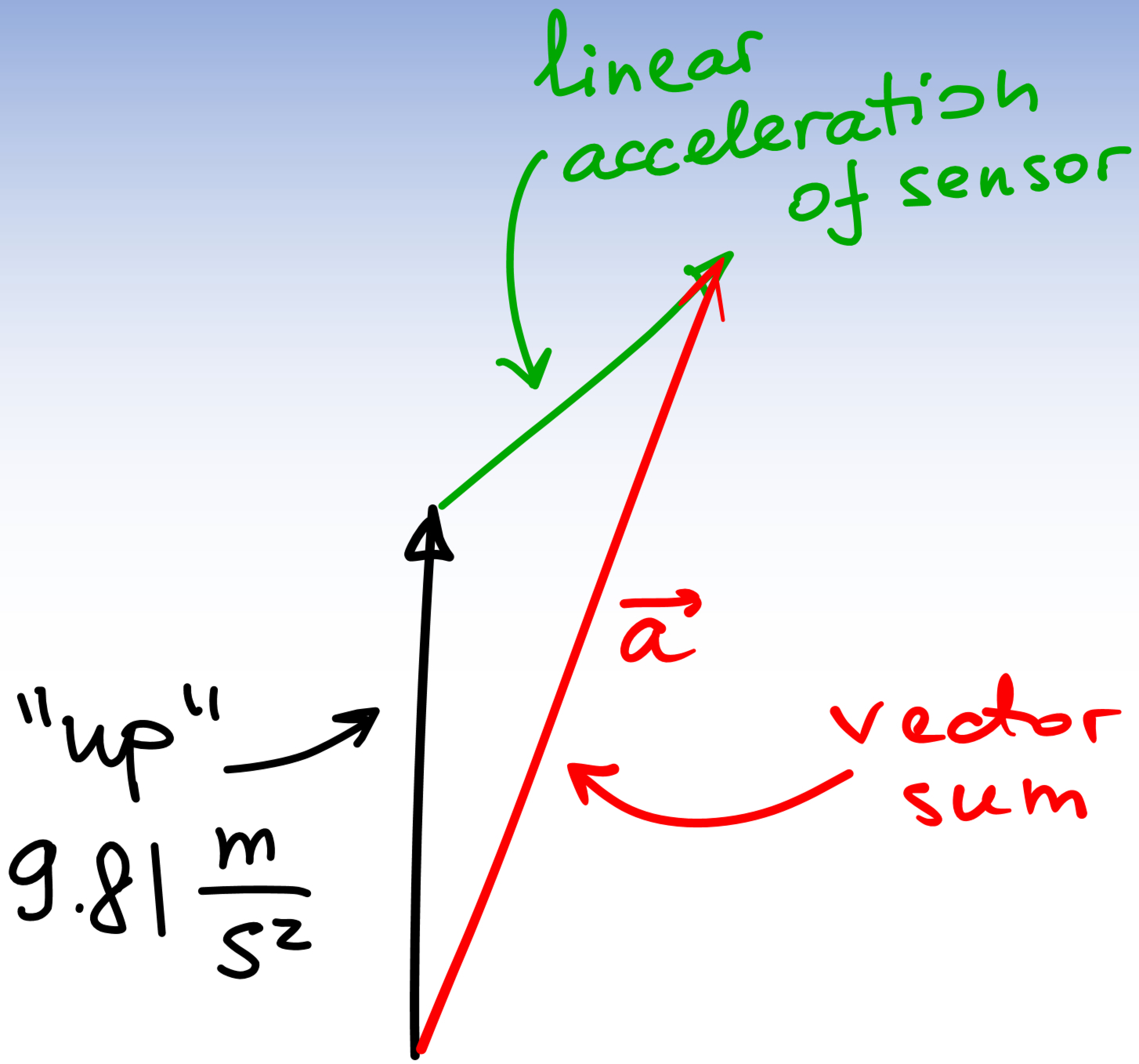






$\vec{a} = (\qquad)$ $\vec{a} = (\qquad)$ $\vec{a} = (\qquad)$

Use Accelerometer to Correct "Tilt Error"



Problem:

Accelerometer measures vector sum of gravity and linear acceleration of sensor.

Solution:

Use heuristic to detect when "not moving" and apply correction only then.

Example: $\|\vec{a}\| \approx 9.81$

Use Magnetometer to Correct Yaw Error

Similar to tilt correction:

- Calculate reference error
- Gradually apply using complementary filter

during each Δt

Problems:

- Vector sum of
- Calibration is
- Field might vary over

accuracy ≈ 5 degrees

Estimating Position and Orientation

Problem setup:

- Allow and track parallax motions
- IMU (gyro + accelerometer + magnetometer) not enough
  - Drift too fast from double integration
  - No way to detect drift errors
- Need sub-millimeter accuracy, stable estimates

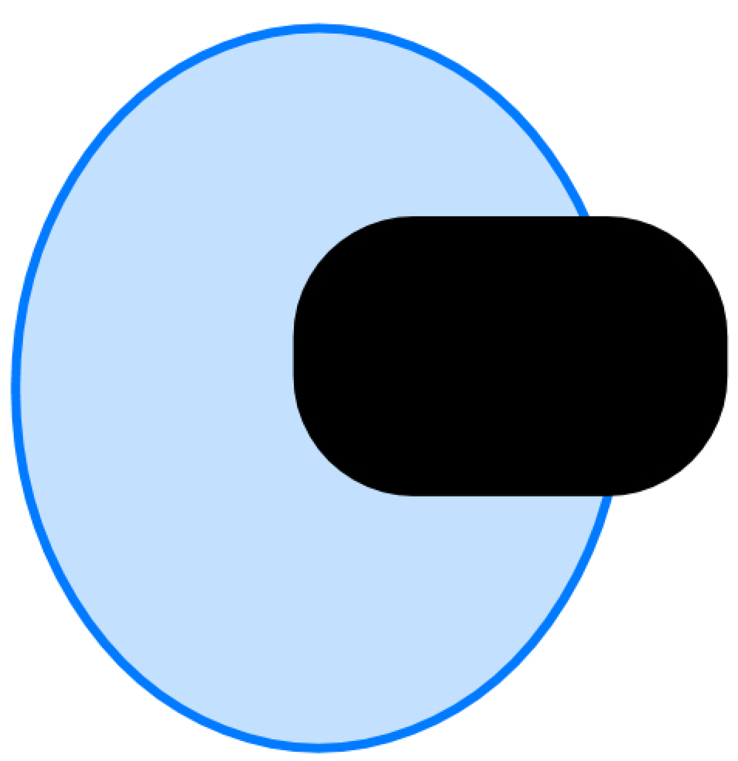
Solutions:

1. Generate non-constant magnetic or EM field
   - RazerHydra, STEM Sixense
   - UWBradio
2. Visibility or line of sight methods

Position Tracking: Visibility Methods

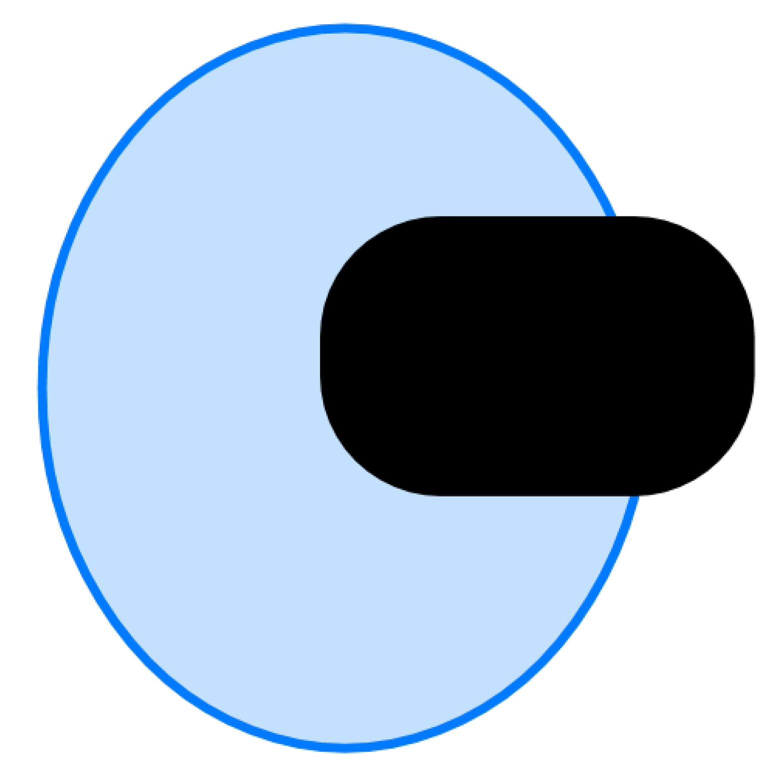
Camera arrangements:

On headset



inside-out

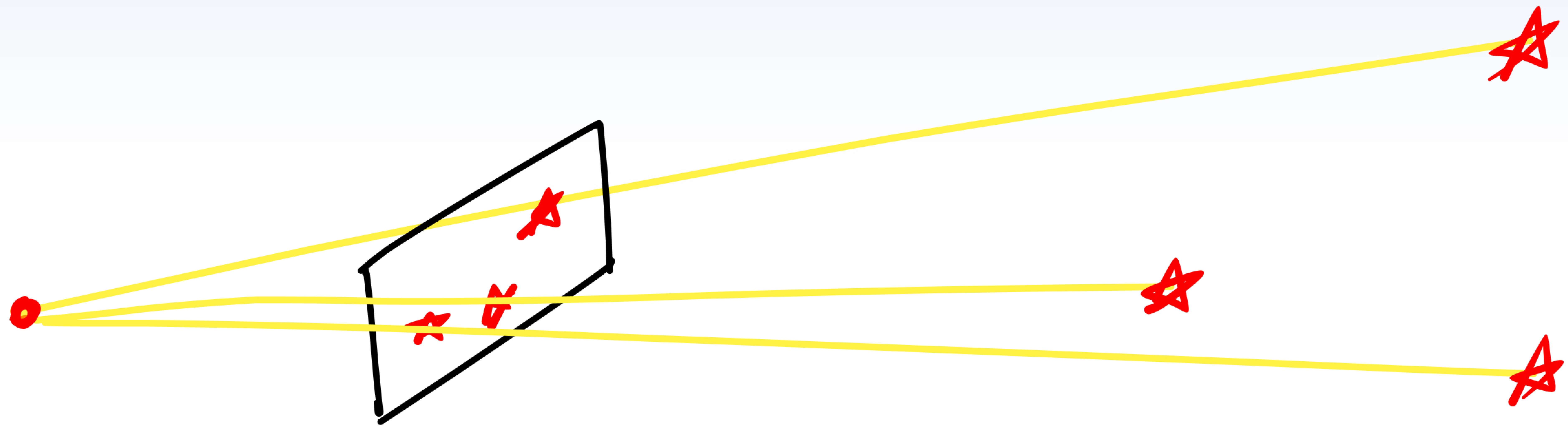
In world



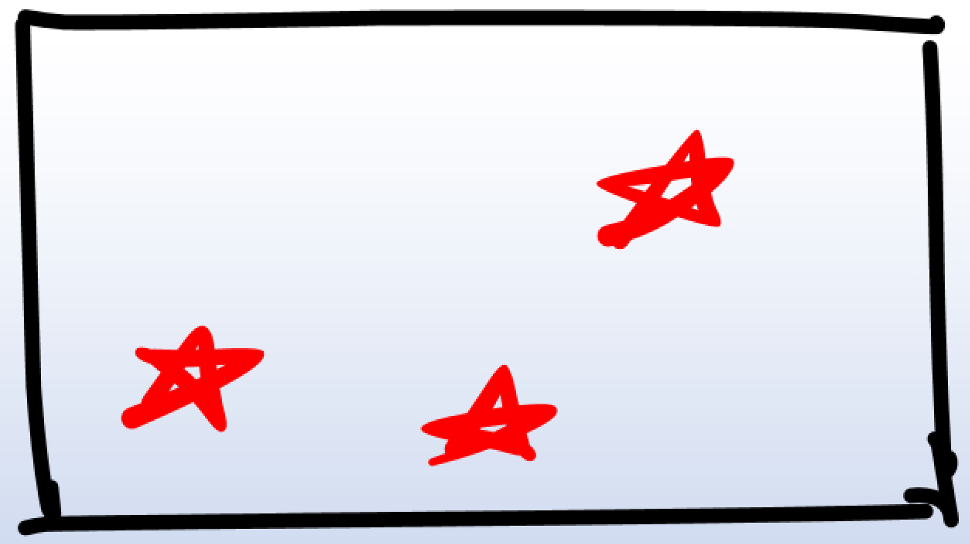
outside-in

Position Tracking: Visibility Methods

Pinhole camera:



Features in an image:



Position Tracking: Visibility Methods

FEATURES:

1) Natural

- Hard computer vision
- Extract and maintain from natural scenes
- Remove moving objects
- Reliability low

2) Artificial

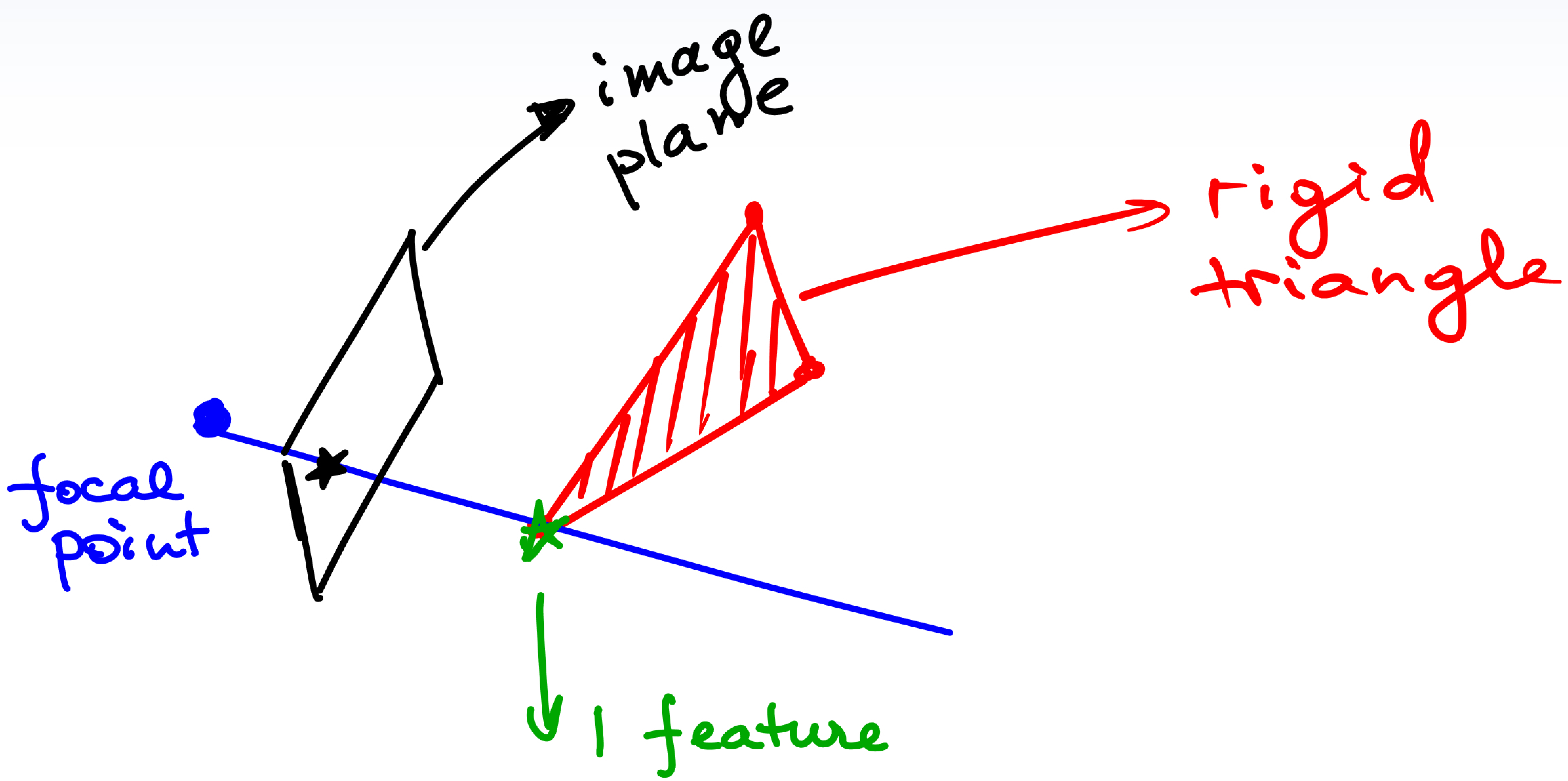
- Trivial computer vision (blob detection)
- QR tags, retro reflective markers, LEDs, laser projections
- Can stay in IR spectrum (invisible to humans)

Position Tracking: Blob Detection

PnP Problem:

Determine rigid body transformation from identified, observed features on a rigid body.

P1P Problem:



DOF analysis:

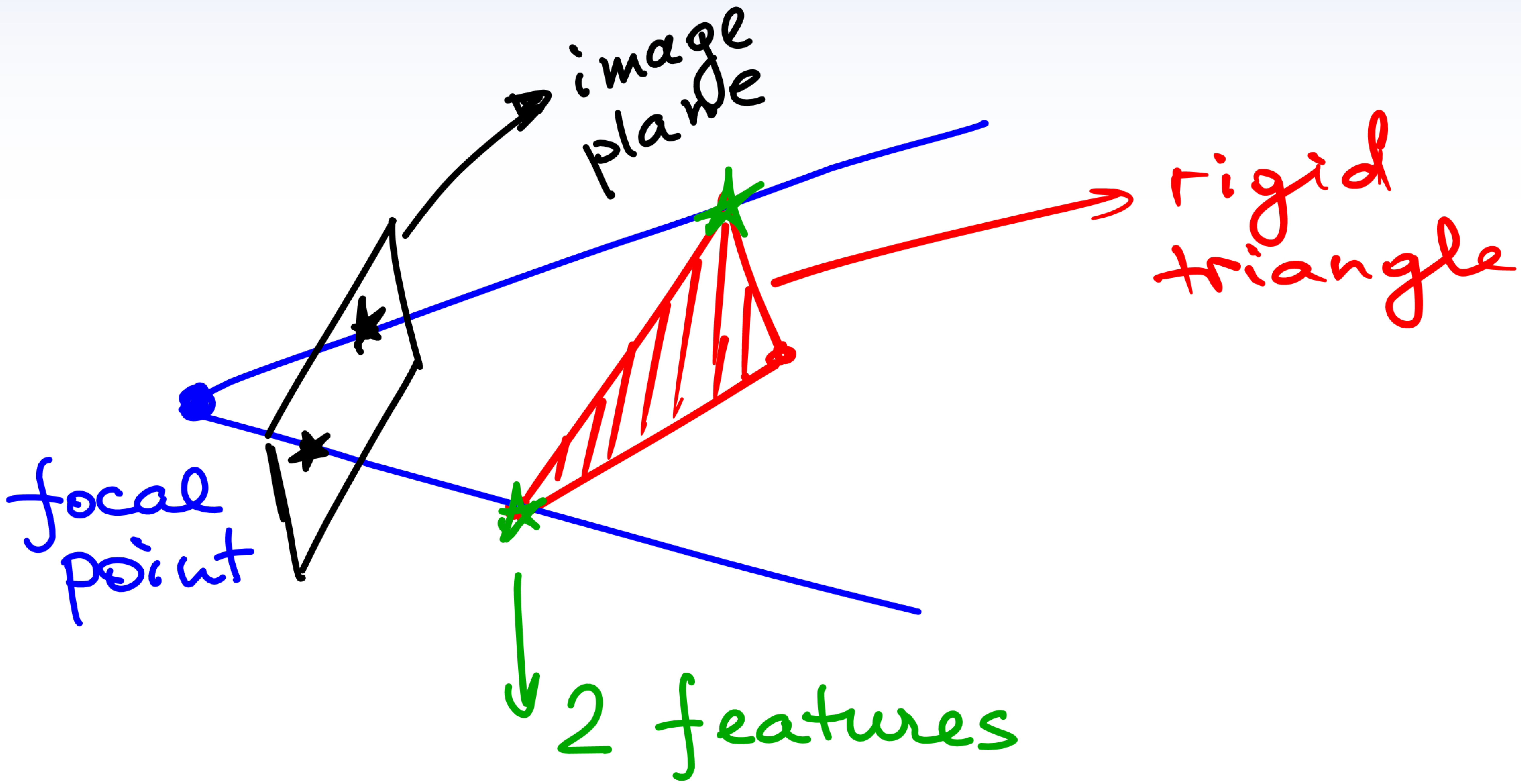
- Start with 6 DOFs (rigid body)
- Each feature subtracts 2 DOFs

DOFs left:

Position Tracking: Blob Detection

P2P Problem:

Determine position and orientation of triangle from features in image.



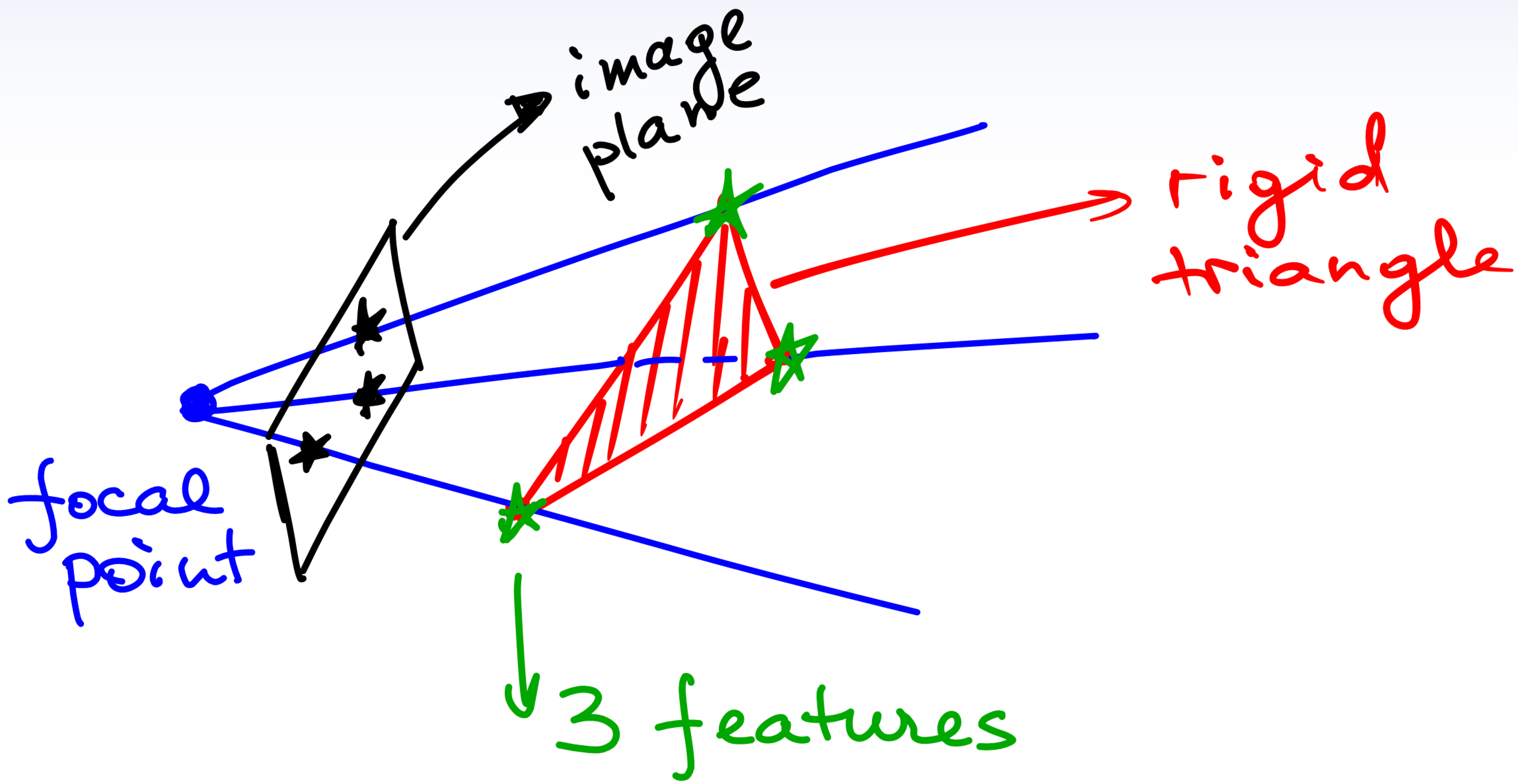
DOFs left:

Position Tracking: Blob Detection

P3P Problem:

Determine position and orientation of triangle from features in image.

DOFs left:



Solution: A system of polynomial equations leads to 8 solutions (but only 4 in front of the camera). The beginnings of computational real algebraic geometry.

Position Tracking: Incremental Blob Detection

Incremental PnP Problem:

Determine position and orientation of triangle from features in image, given current estimate.