香港科技大學 THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY | 電子及計算機工程學系 DEPARTMENT OF ELECTRONIC & COMPUTER ENGINEERING

Introduction to Aerial Robotics
Lecture 1

Shaojie Shen

Assistant Professor

Dept. of ECE, HKUST

1 September 2015

Logistics

About this Course…

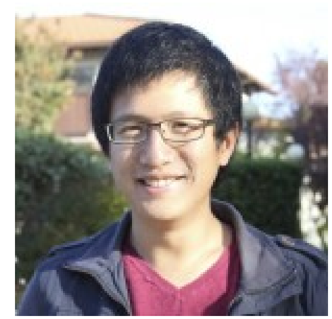
- This course is about:
  - Autonomy for aerial robots
  - Modeling & control of multi-rotor aerial robots
  - Algorithms for vision-based state estimation
  - Algorithms for multi-sensor fusion
  - Real-time implementation of software modules for autonomous flight of aerial robots
  - Lots of fun with robots ☺
- This course is NOT about:
  - Aerodynamics ☹
  - Aircraft design ☹
  - Electromechanical systems of aerial robots ☹

Course structure

- Project-based, with 1 in-class midterm exam
- Projects comprise:
  - Take home programming tasks
  - Tutorials
  - Lab sessions
- Grading:
  - Midterm: 30%
  - Project 1: 20%
  - Project 2: 20%
  - Project 3: 30%

Teaching Team

- **Instructor:**
  - Shaojie Shen (eeshaojie@ust.hk)
  - Office Hour: Tuesday 4:30pm-6:00pm or by appointment



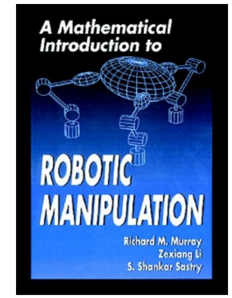
- **Teaching Assistants:**
  - Kunyue Su (ksu@connect.ust.hk)
  - Zhenfei Yang (zyangag@connect.ust.hk)
  - Office Hour: TBD

Administrative Stuff

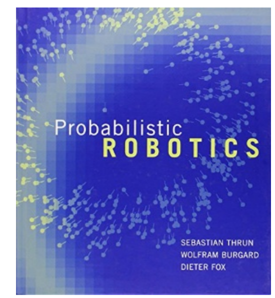
- **Lecture:**
  - Tuesday 1:30pm - 4:20pm, G002 CYT Bldg
- **Labs:**
  - LA1: Wednesday 6:00pm - 8:50pm, Rm2014 CYT Bldg
  - LA2: Thursday 2:00pm - 4:50pm, Rm2014 CYT Bldg
- **Course Email:**
  - elec6910p@gmail.com
- **Course Website:**
  - http://course.ee.ust.hk/elec6910p/

Reference Books

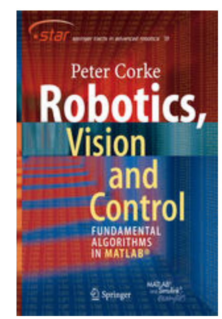
- A Mathematical Introduction to Robotic Manipulation;
  - Richard M. Murray, Zexiang Li, S. Shankar Sastry;



- Probabilistic Robotics;
  - Sebastian Thrun, Wolfram Burgard, Dieter Fox;



- Robotics, Vision and Control: Fundamental Algorithms in MATLAB;
  - Peter Corke;



Workload & Expectation

- **Expected Student Background:**
  - Linear algebra
  - Probability
  - MATLAB programming skills
  - C++ programming skills (VERY IMPORTANT)
  - Linux
  - Love robots ☺

- **Workload:**
  - Attend lectures & labs
  - Lots of project work
  - Have fun with robots ☺

Programming!

- MATLAB
- C++
- Robot Operating System (ROS) [http://www.ros.org/]
- Requires coding on both desktop PC and embedded platforms

ROS

Schedule

- Dynamics & Control
- Vison
- Estimation

Week	Lecture Date	Topic	Lab	Assignment
1	1/9	Introduction Rigid Body Transformation	x	
2	8/9	Rigid Body Transformation Quadrotor Modeling	x	Project 1 Out
3	15/9	Control Basics Quadrotor Control	x	
4	22/9	Time & Motion Trajectory Generation	ROS Tutorial 1	
5	29/9	Camera Modeling & Calibration Feature Detection & Matching	x	
6	6/10	Optical Flow Dense Stereo	Project 1 Phase 3	
7	13/10	Multi-View Geometry 2D-2D, 3D-2D, 3D-3D	Project 1 Phase 3	Project 1 Due Project 2 Out
8	20/10	**Midterm Exam**	x	
9	27/10	Probability Basics Bayesian Inferencing Maximum Likelihood Estimation	ROS Tutorial 2	
10	3/11	Kalman Filter Sensor Fusion	x	Project 2 Due Project 3 Out
11	10/11	SLAM	Project 3 Phase 2	
12	17/11	No Lecture	Project 3 Phase 2	
13	24/11	No Lecture	Project 3 Phase 2	Project 3 Due

Overview of Aerial Robotics

Aerial Robots = Military Drones?

- Mostly Big
- 10+ Hours of Flight Time
- Remote Control
  - Waypoint planning
  - Some joysticking
  - Flight crew 4-10

Drones mischaracterize what these things are. They're not dumb. Nor are they unmanned, actually. They're remotely piloted aircraft. - Gen. Norton Schwarz, August 10, 2012

Unmanned Aerial Vehicles

- Projected civilian market of $400 billion USD by 2020
- Regulations are coming in place

Micro Aerial Vehicles

- Small size (<1m)
- Adequate payload (1-5kg)
- Low cost (< 10k USD)
- Safe
- Superior mobility


Inspection


Search and Rescue


Transportation

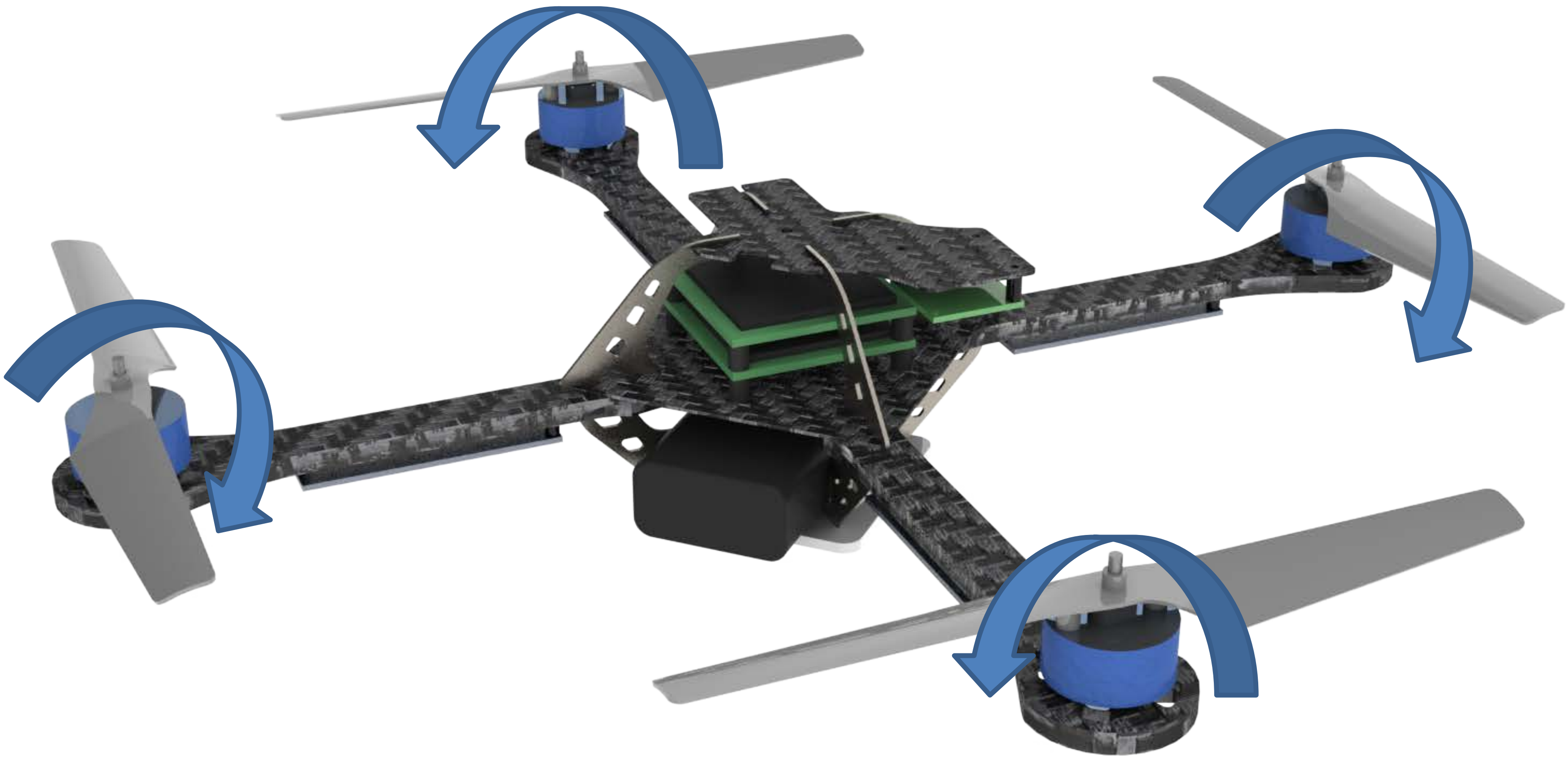

Aerial Photography

Law enforcement


Agriculture

Commercialization of Aerial Robots

Quadrotors

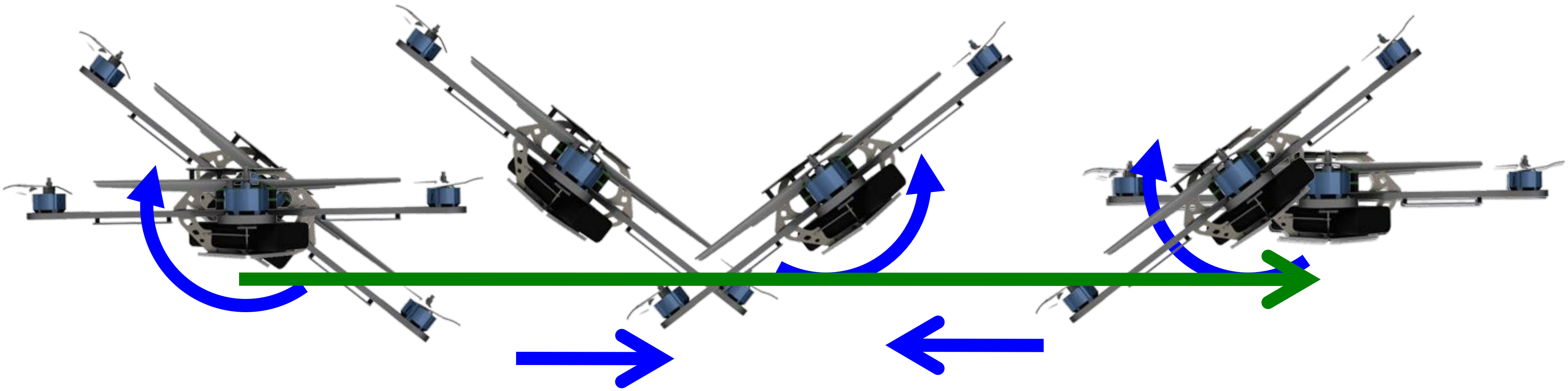


CAD Models from V. Kumar

Quadrotors - Rotation

CAD Models from V. Kumar

Quadrotors - Translation



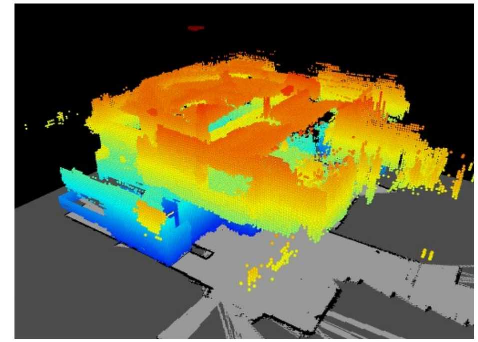
CAD Models from V. Kumar

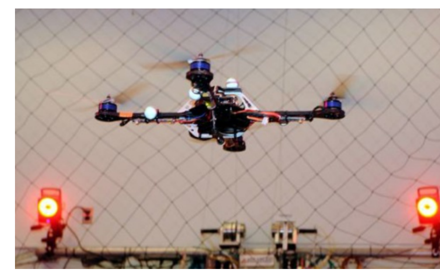
Towards Autonomous Flight…

- Sensing and perception — Part 2
- State estimation — Part 3
- Planning & obstacle avoidance
- Trajectory generation
- Control

Part 1 (Trajectory generation, Control)

- System integration — Projects!





Towards Autonomous Flight…

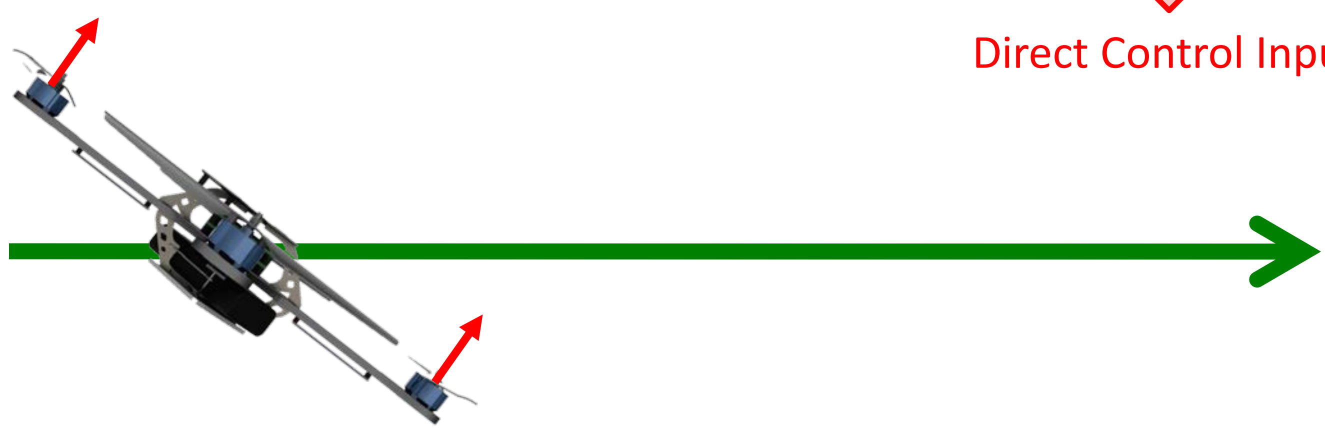
- Dynamics & Control
- Vison
- Estimation

Week	Lecture Date	Topic	Lab	Assignment
1	1/9	Introduction Rigid Body Transformation	x	
2	8/9	Rigid Body Transformation Quadrotor Modeling	x	Project 1 Out
3	15/9	Control Basics Quadrotor Control	x	
4	22/9	Time & Motion Trajectory Generation	ROS Tutorial 1	
5	29/9	Camera Modeling & Calibration Feature Detection & Matching	x	
6	6/10	Optical Flow Dense Stereo	Project 1 Phase 3	
7	13/10	Multi-View Geometry 2D-2D, 3D-2D, 3D-3D	Project 1 Phase 3	Project 1 Due Project 2 Out
8	20/10	**Midterm Exam**	x	
9	27/10	Probability Basics Bayesian Inferencing Maximum Likelihood Estimation	ROS Tutorial 2	
10	3/11	Kalman Filter Sensor Fusion	x	Project 2 Due Project 3 Out
11	10/11	SLAM	Project 3 Phase 2	
12	17/11	No Lecture	Project 3 Phase 2	
13	24/11	No Lecture	Project 3 Phase 2	Project 3 Due

How to fly? – Dynamics & Control

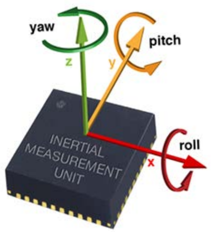
Derivative	Translation	Rotation	Thrust
0	Position	Control Target	
1	Velocity		
2	Acceleration	Rotation	
3	Jerk	Angular Velocity	
4	Snap	Angular Acceleration	Differential Thrust
5	Crackle	Angular Jerk	Change in Thrust

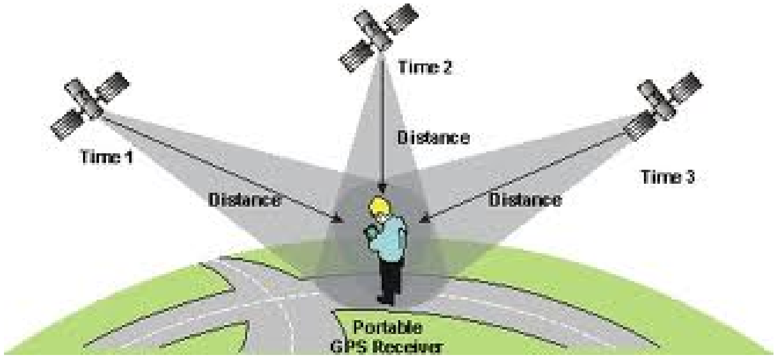
Direct Control Input



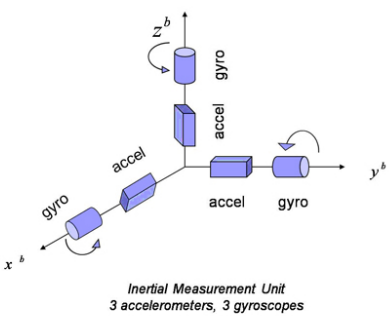
How to Fly? – Sensing & Estimation

- Answer the key question: Where am I?
- Proprioceptive sensors
  - Low cost Inertial Measurement Unit
- Exteroceptive sensors
  - GPS
  - Magnetometer
  - Barometer
  - Cameras
  - Laser range finders
  - etc.
- Processors
- Algorithms

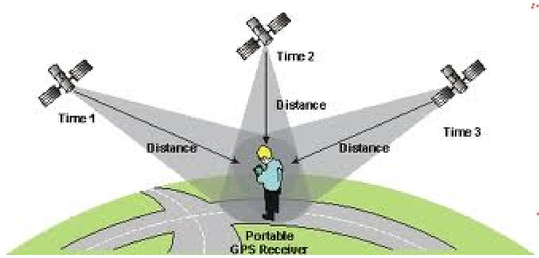




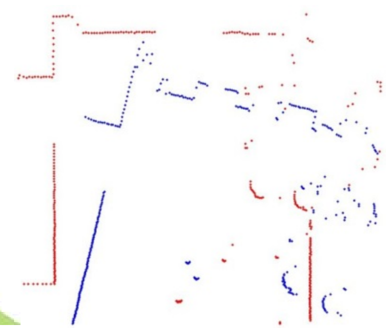
How to Fly? – Sensing & Estimation



Inertial Measurement Unit



GPS

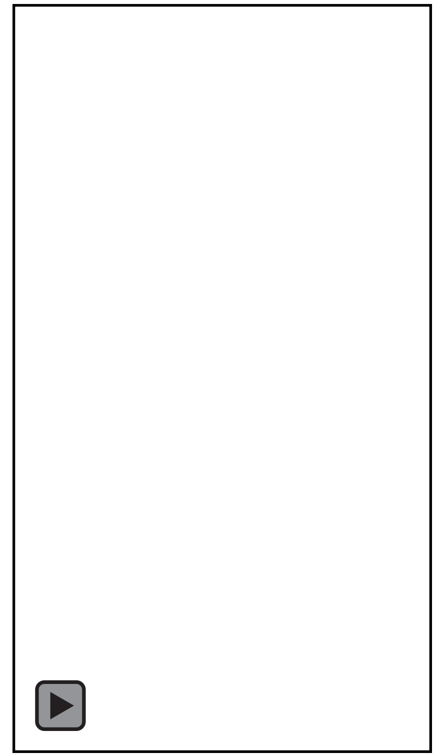


Laser Range Finder

Camera

How to Fly? – Navigation

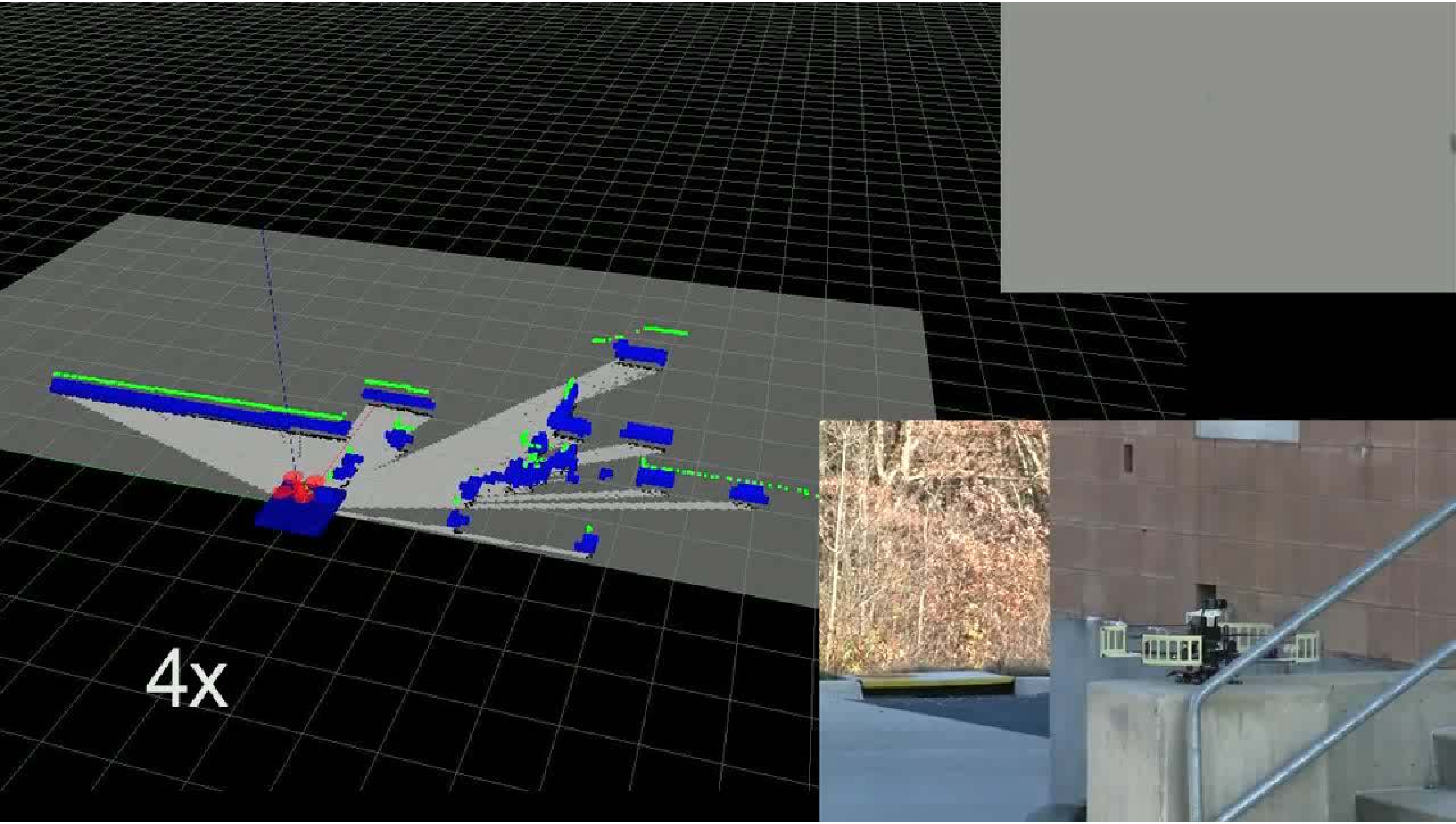
- Remote control
  - Requires line of sight
  - Requires communication link
  - Requires skilled pilots
- Inertial navigation
  - Requires aviation grade IMU
  - Heavy and expensive
- GPS-based navigation
  - Waypoint following
  - No obstacle avoidance
  - GPS can be unreliable



How to Fly? – GPS-based Navigation

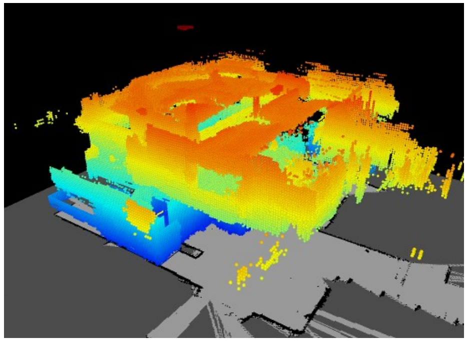
How to Fly? – GPS-denied Navigation

How to Fly? – Laser-based Navigation

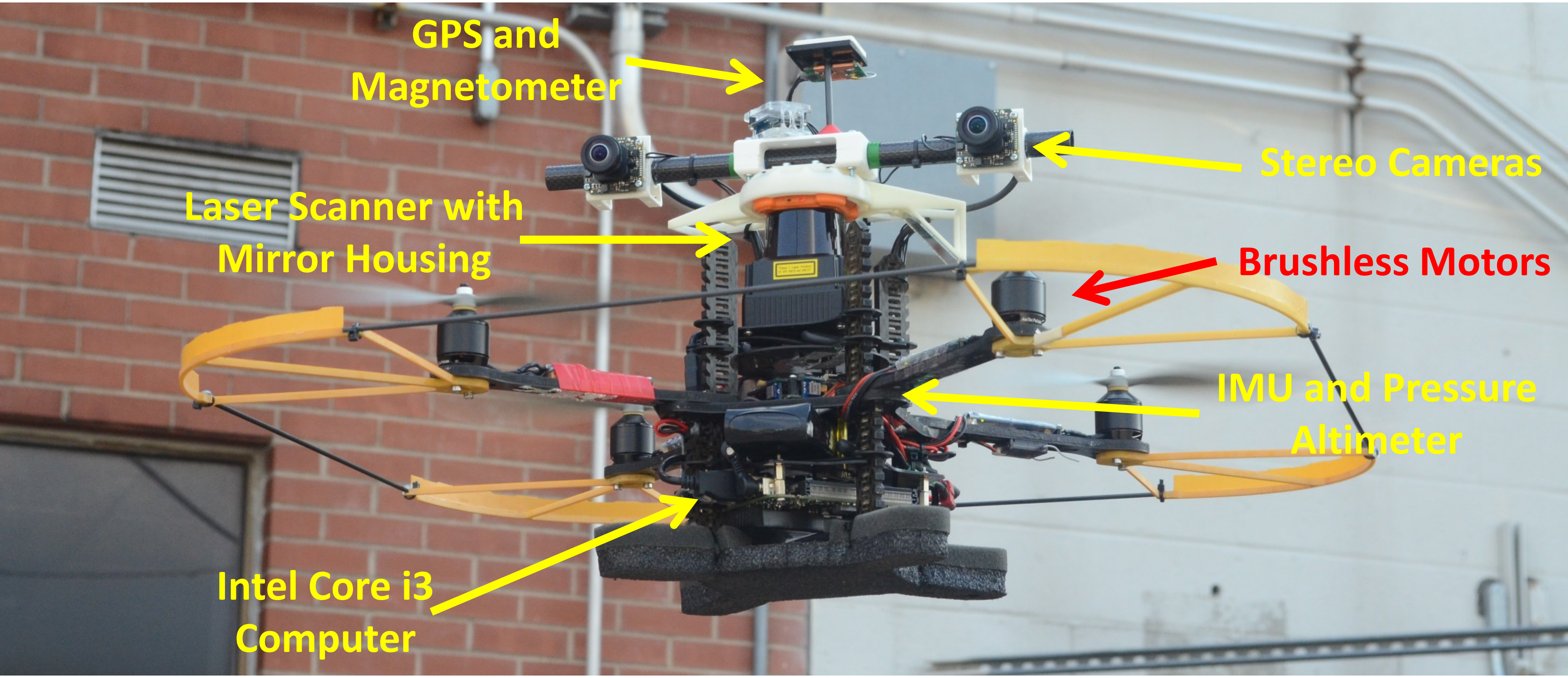


Challenges

- Sensing & Perception
  - 3D sensing & mapping
- State Estimation & Localization
  - Low latency & high accuracy
- Obstacle avoidance
  - Complex & unknown environments
- Trajectory Control
  - Aggressive maneuvers
  - Smooth trajectory tracking
- System integration
  - Limited sensing & computation
  - Autonomous operations



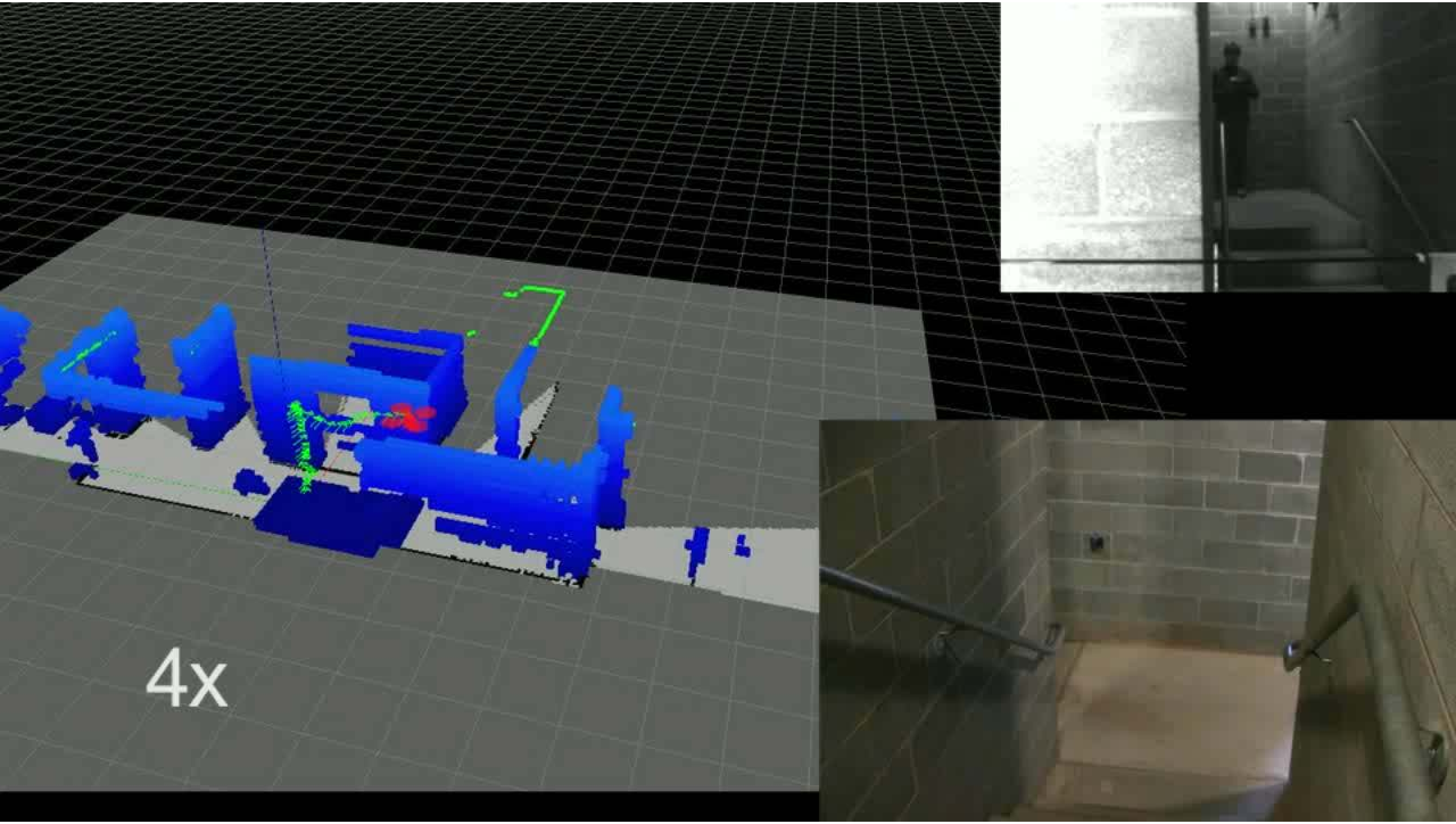
A Flying Robot



Goal

From Movie “Prometheus”

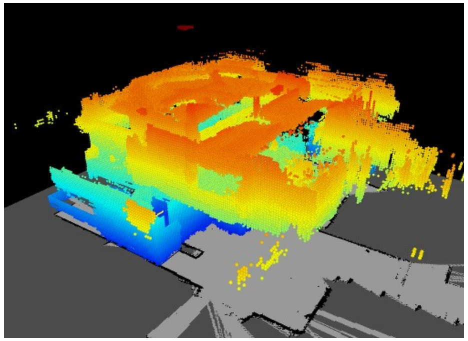
Fly over Multiple Floors

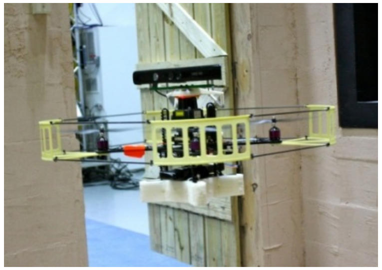


Shen, et al, 2013

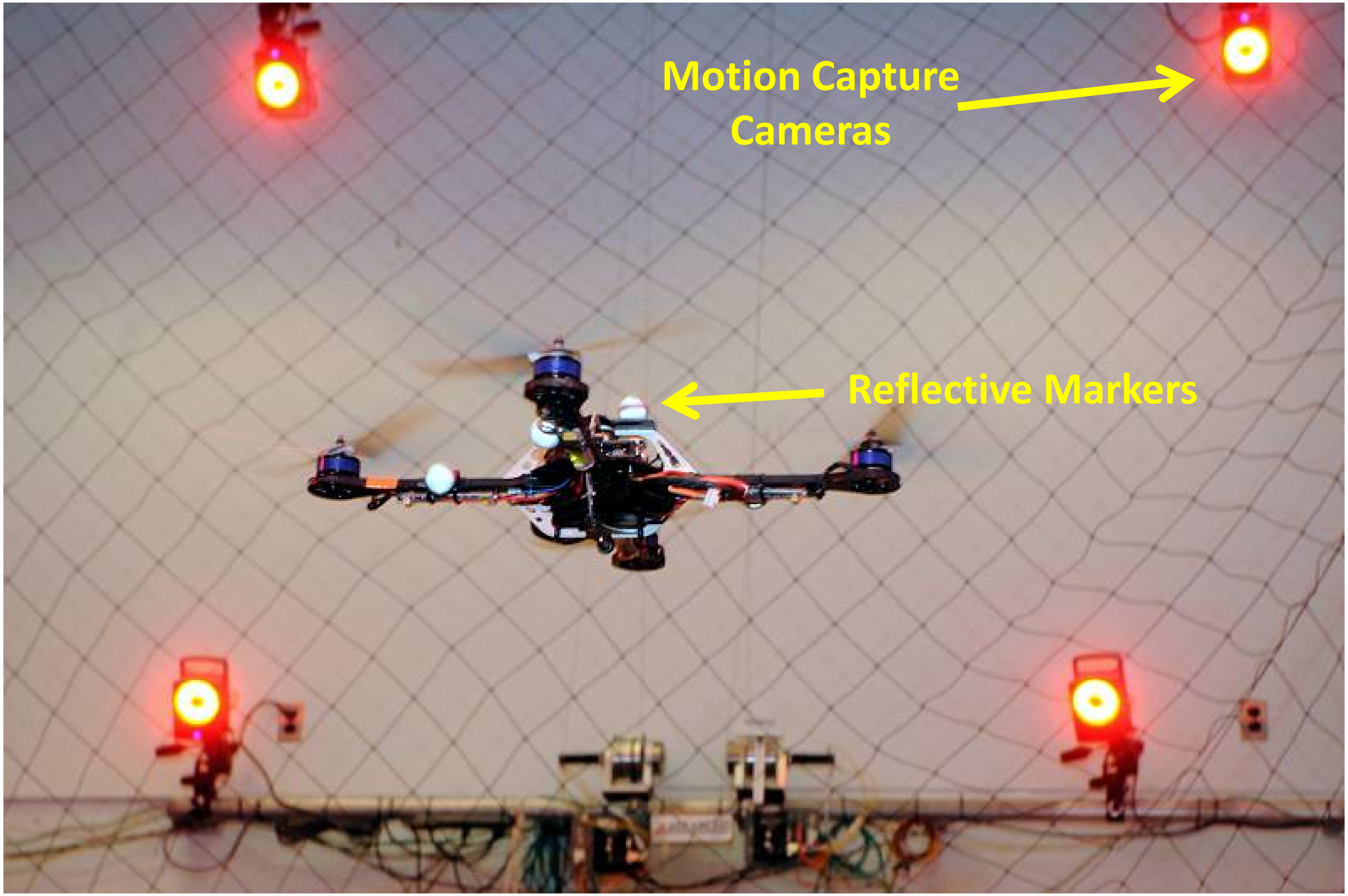
Towards Autonomous Flight…

- Sensing and perception
- State estimation
- Planning & obstacle avoidance
- Trajectory generation
- Control
- System integration





Bypass the Sensing Problem



Robust Control

Mellinger, 2011

Flying Inverted Pendulum

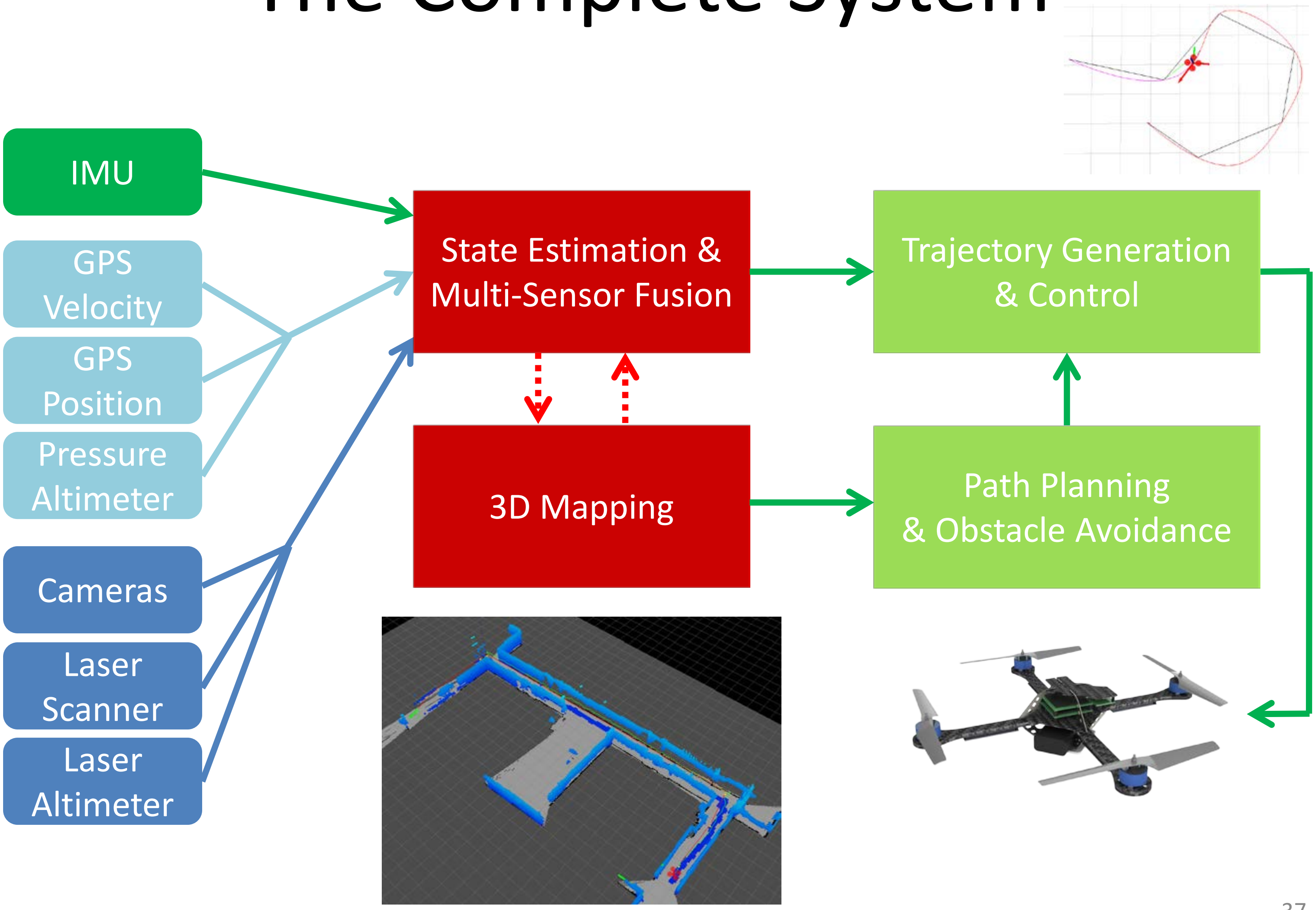
A Flying Inverted Pendulum

IDSC

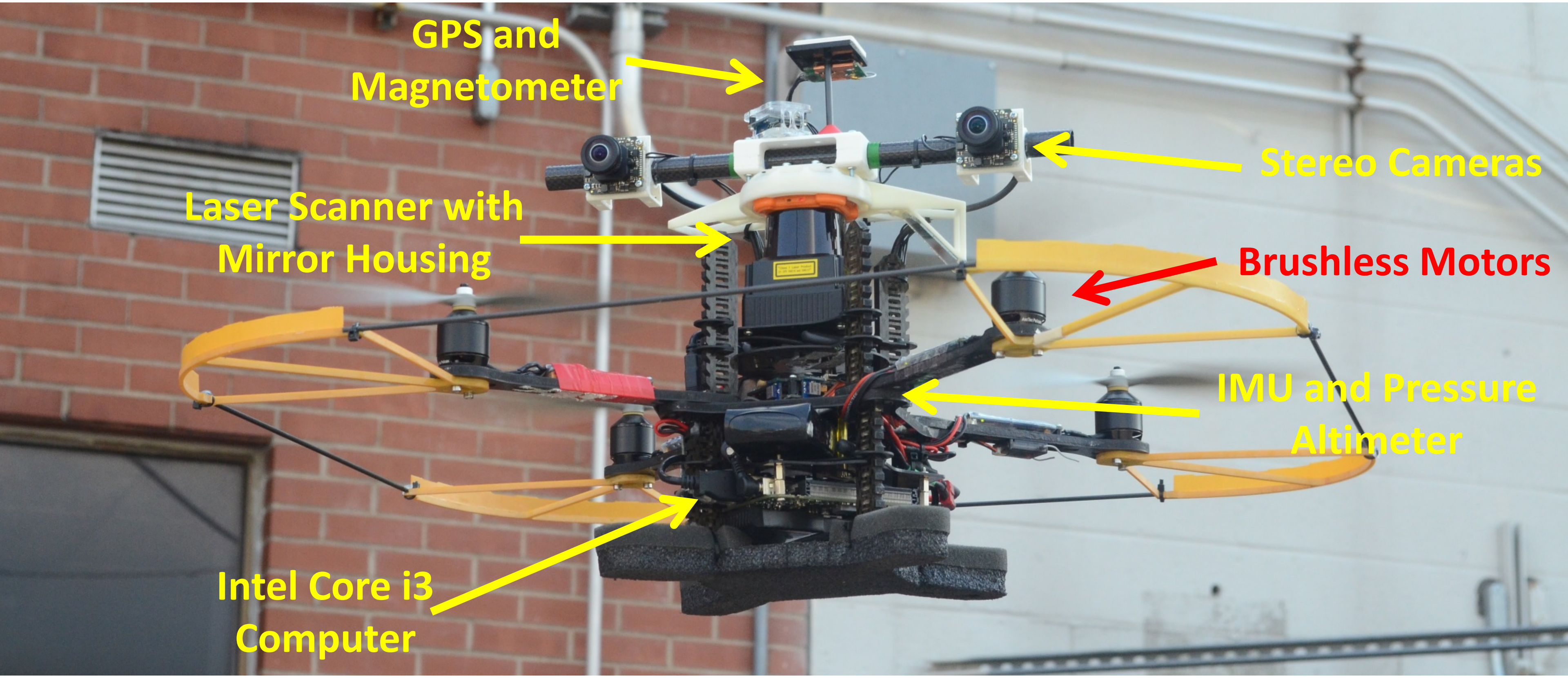
ETH Zürich

Cooperative Ball Throwing

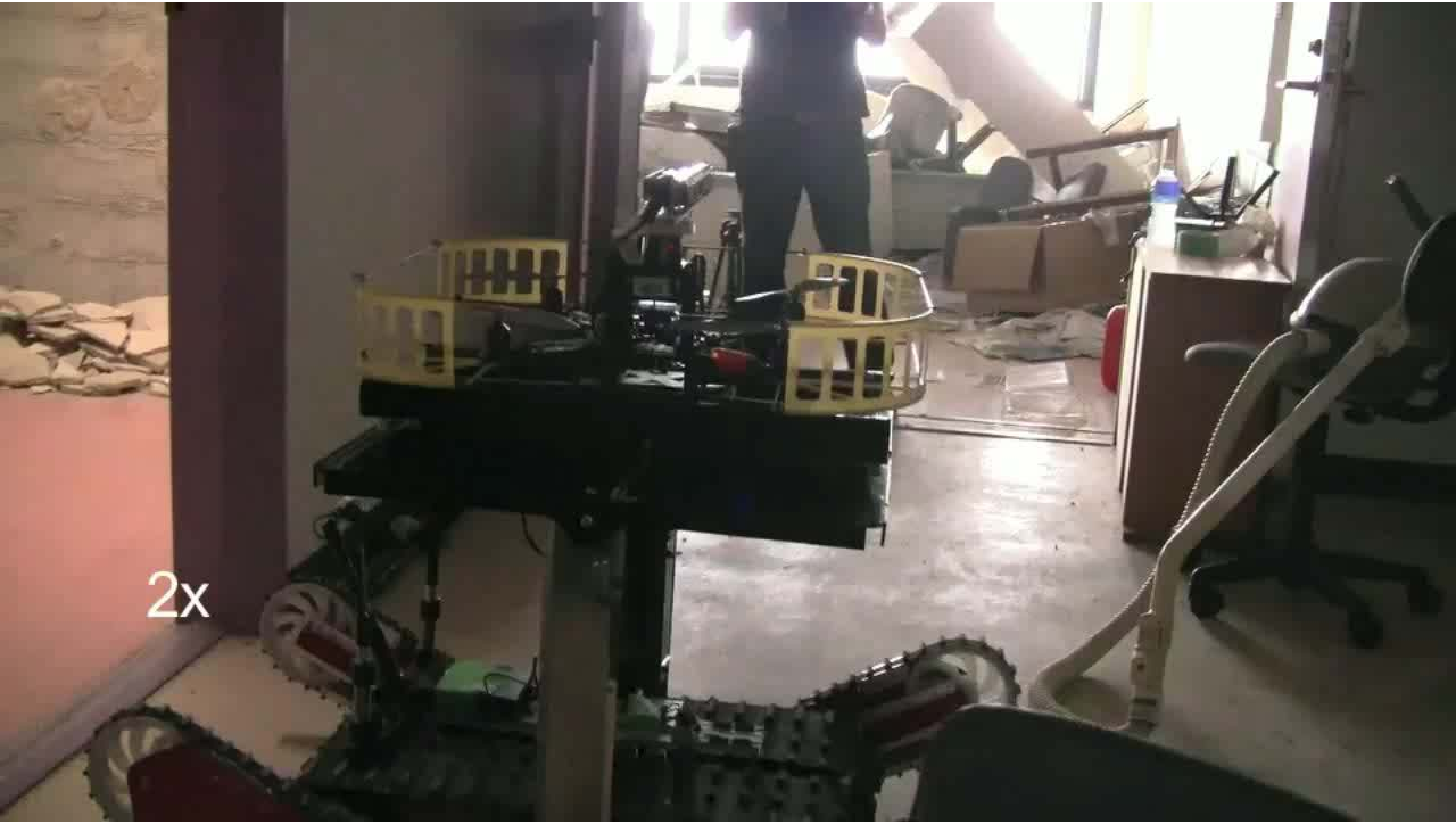
The Complete System



A Flying Robot



Search and Rescue

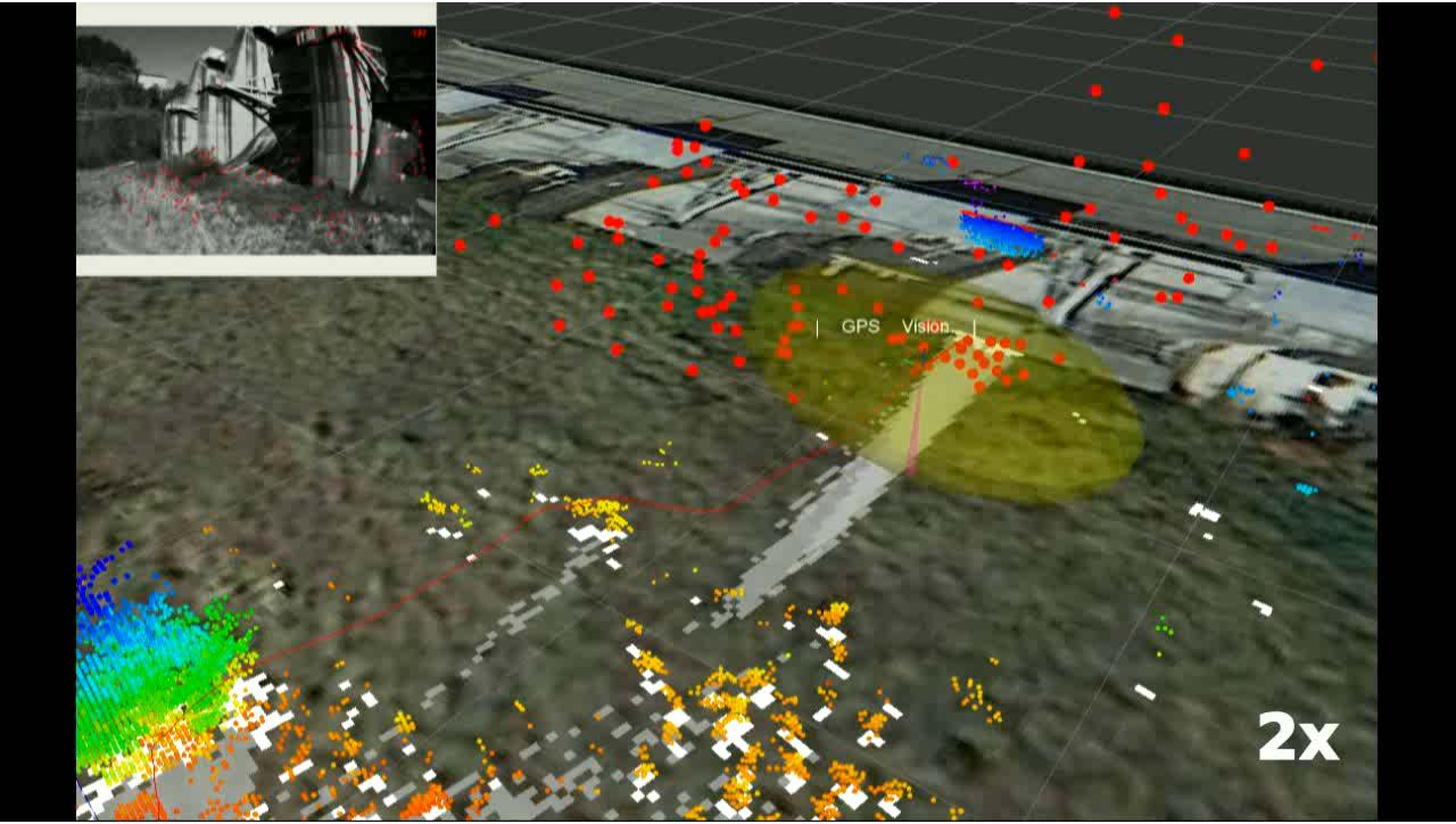


Michael, et al, 2012

Dam Inspection

Emergency flooding gates at the Carter's Dam, GA, USA

Dam Inspection



Shen, et al, 2014

Dam Inspection

Penstock at the Carter’s Dam, GA, USA

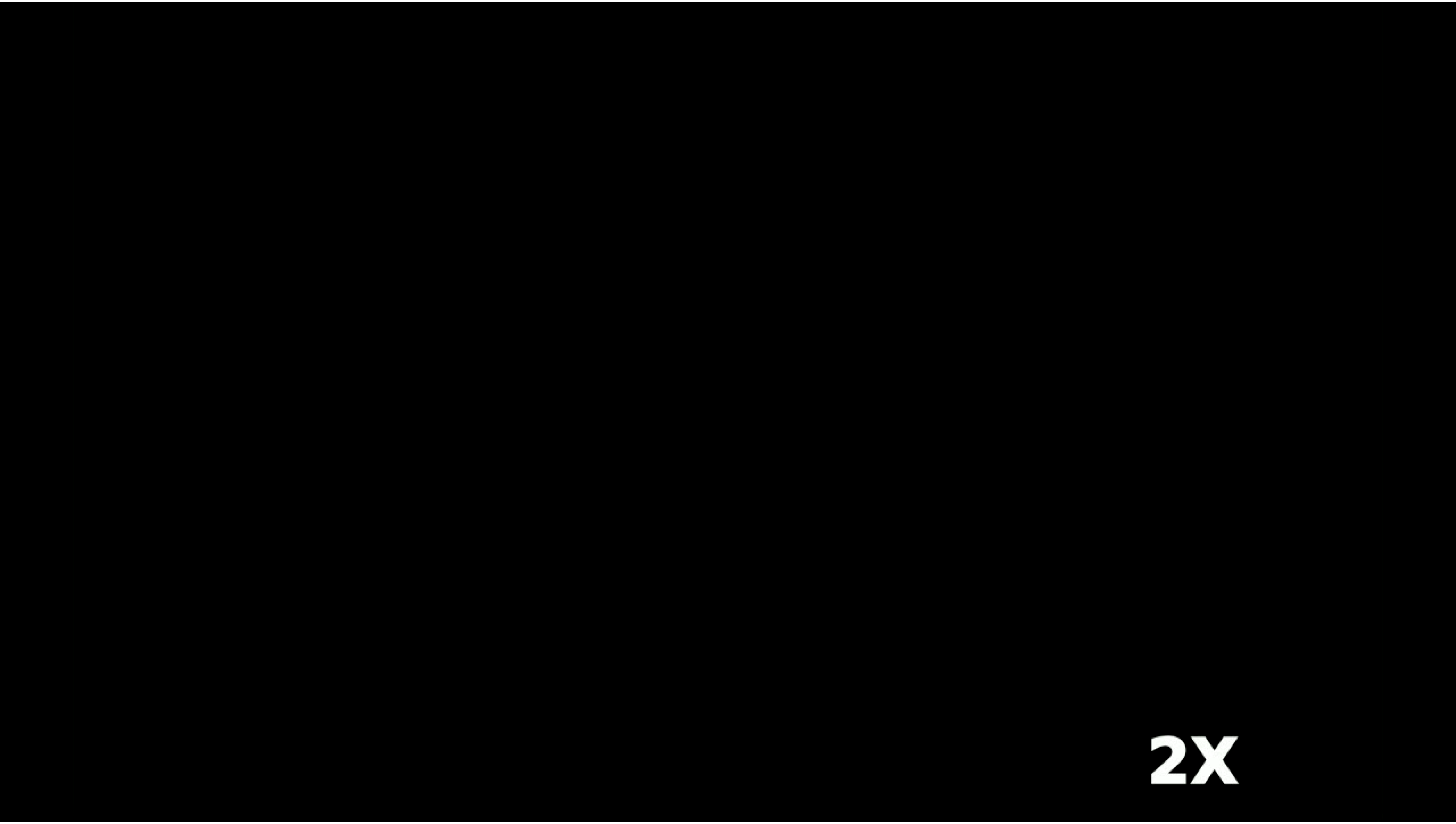
Dam Inspection

Ozaslan, et al, 2014

Precision Farming

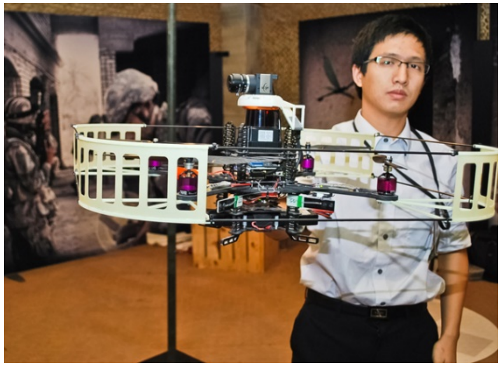
E.&J. Gallo Winery, CA, USA

Precision Farming



Shen, et al, 2014

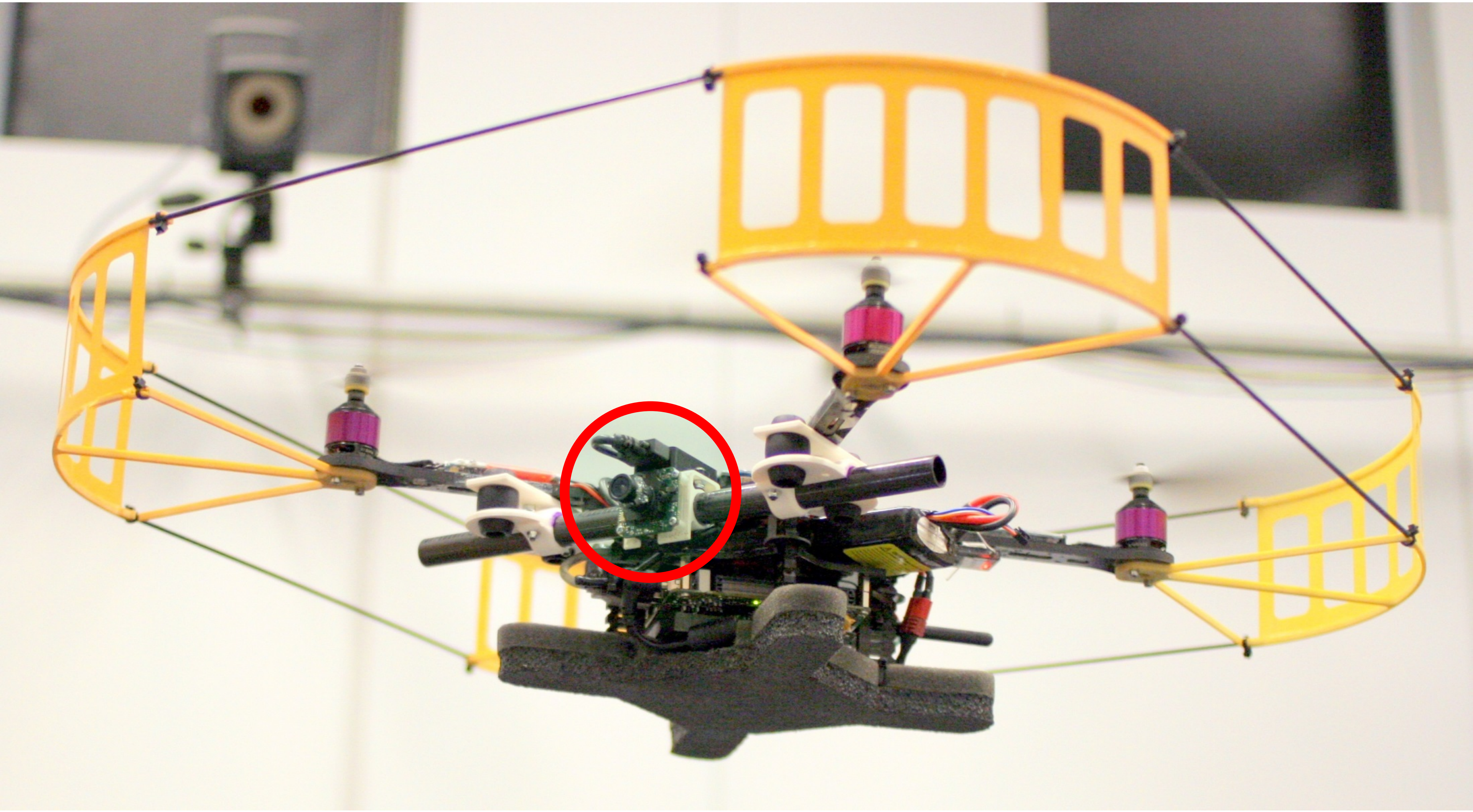
Small and Autonomous



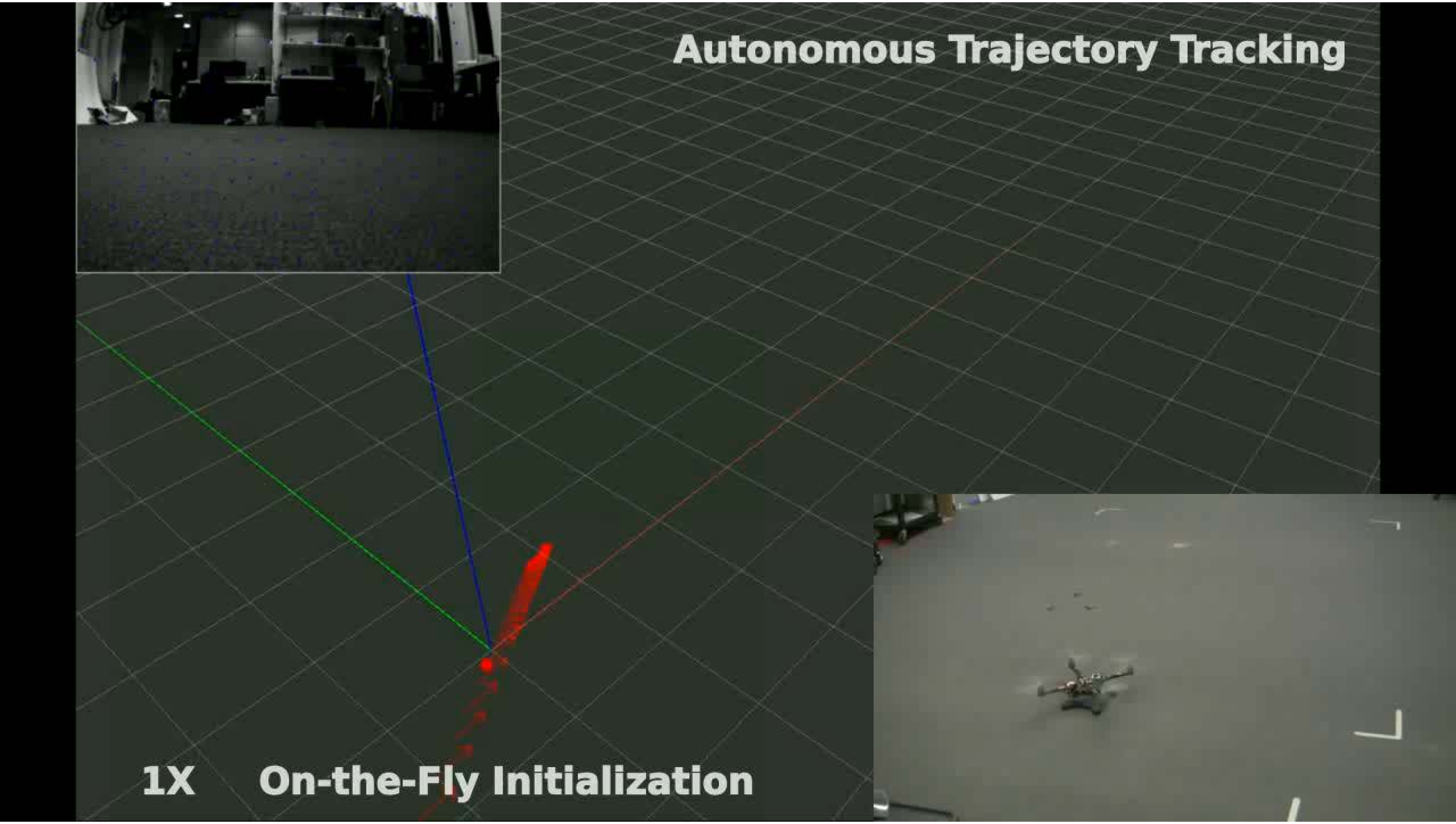
Processing: Not quite there yet, but remember the Moore's Law!

Sensing: YES!

Minimum Sensing: 1 Camera + 1 IMU

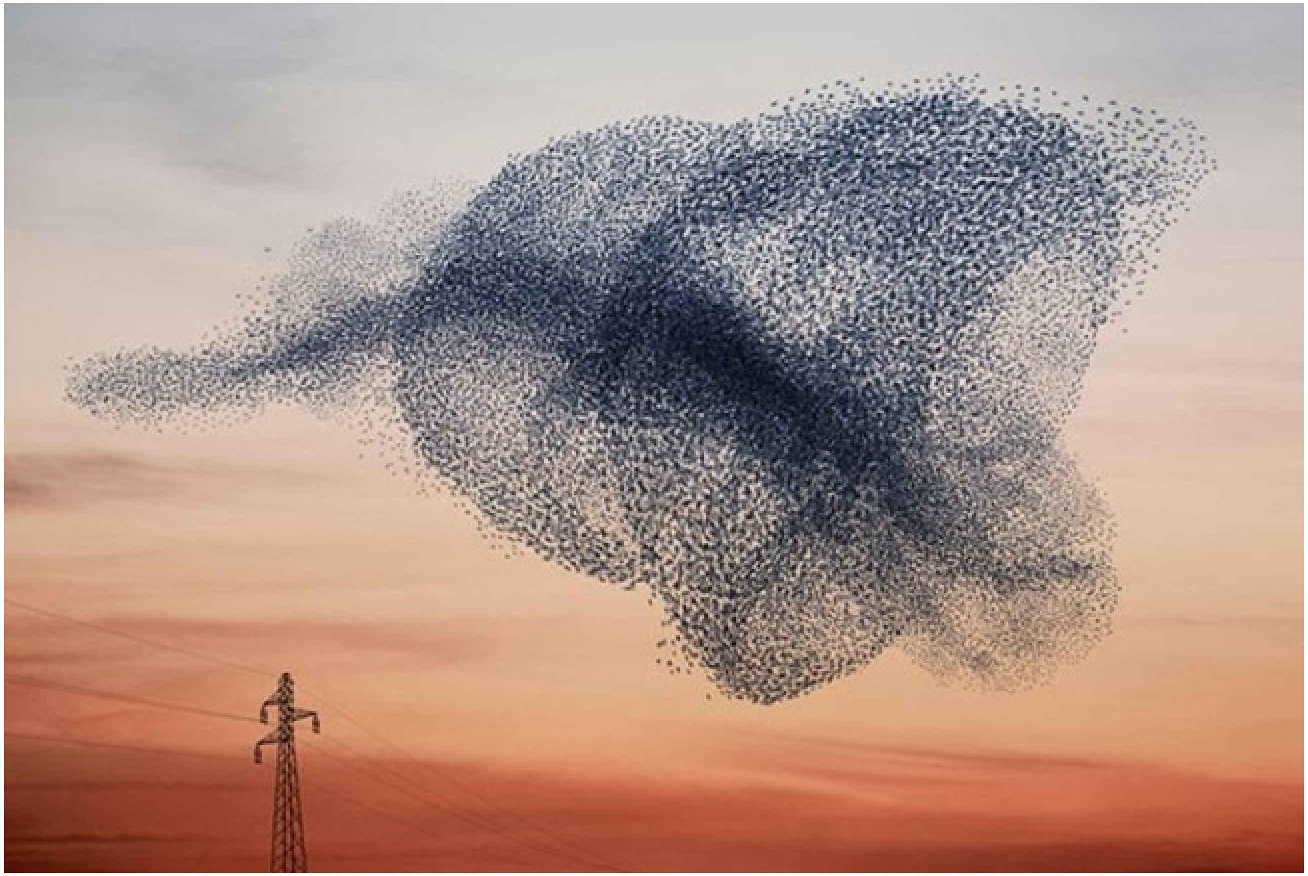


Minimum Sensing: 1 Camera + 1 IMU

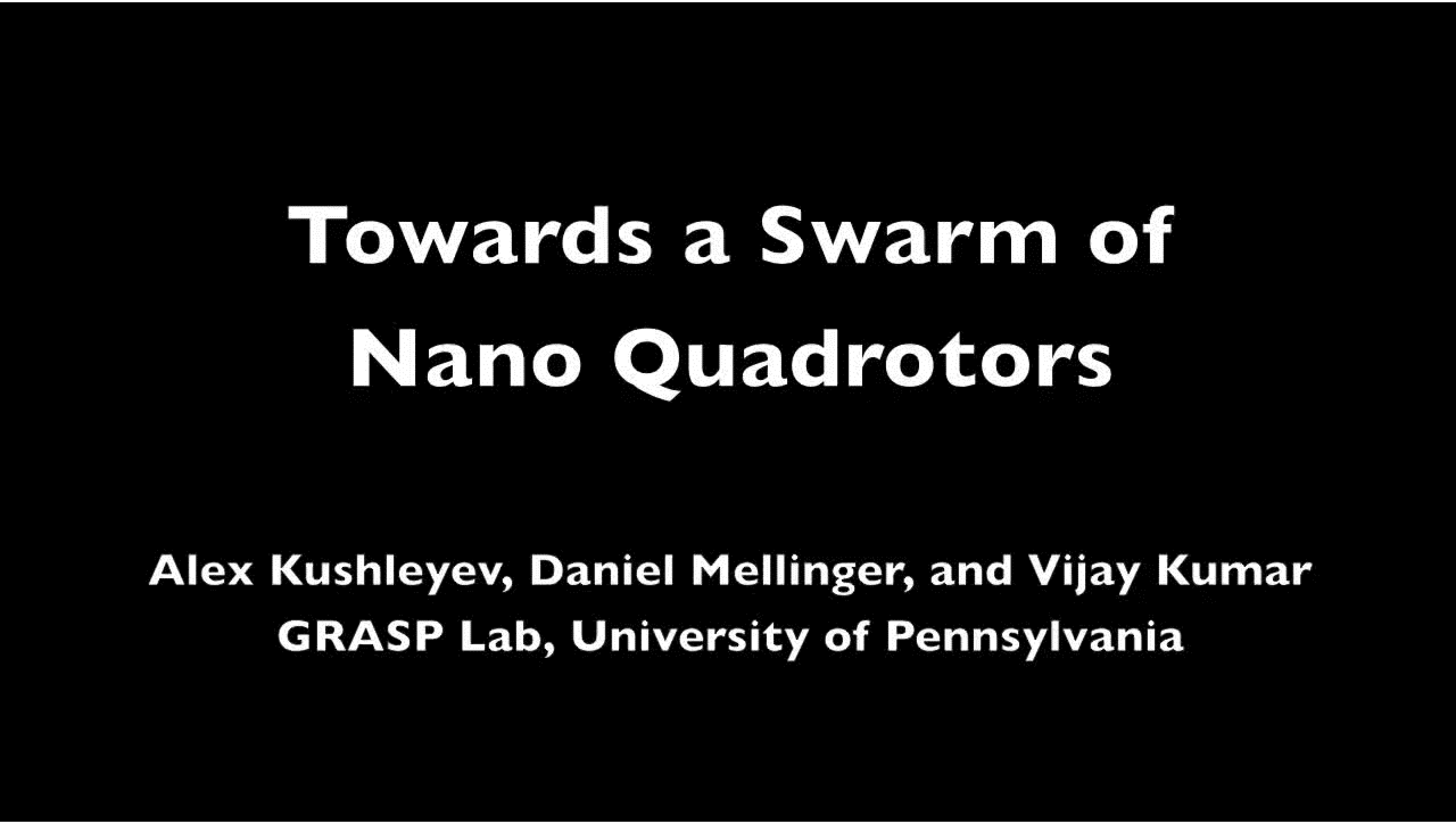


Shen, et al, 2014

Swarm!



Swarm!



Mellinger, et al, 2013

Amazing In Motion

Kmel Robotics, 2014

Course Outline

Course Outline

- Dynamics & Control
- Vison
- Estimation

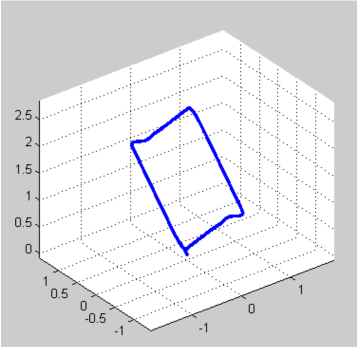
Week	Lecture Date	Topic	Lab	Assignment
1	1/9	Introduction Rigid Body Transformation	x	
2	8/9	Rigid Body Transformation Quadrotor Modeling	x	Project 1 Out
3	15/9	Control Basics Quadrotor Control	x	
4	22/9	Time & Motion Trajectory Generation	ROS Tutorial 1	
5	29/9	Camera Modeling & Calibration Feature Detection & Matching	x	
6	6/10	Optical Flow Dense Stereo	Project 1 Phase 3	
7	13/10	Multi-View Geometry 2D-2D, 3D-2D, 3D-3D	Project 1 Phase 3	Project 1 Due Project 2 Out
8	20/10	**Midterm Exam**	x	
9	27/10	Probability Basics Bayesian Inferencing Maximum Likelihood Estimation	ROS Tutorial 2	
10	3/11	Kalman Filter Sensor Fusion	x	Project 2 Due Project 3 Out
11	10/11	SLAM	Project 3 Phase 2	
12	17/11	No Lecture	Project 3 Phase 2	
13	24/11	No Lecture	Project 3 Phase 2	Project 3 Due

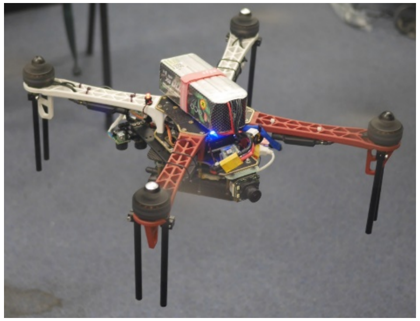
Course Outline

- Dynamics & Control
  - Understand 3D rigid body transformation
  - Derive the model of quadrotor aerial robots
  - Able to hover the quadrotor given perfect localization
  - Able to generate and track smooth trajectories given user-defined waypoints
  - Project 1
- Vison
  - Understand how cameras capture the world
  - Understand standard feature-based image processing pipeline
  - Able to use only images to compute camera position and orientation
  - Project 2
- Estimation
  - Understand how to incorporate probability principles into robotics
  - Able to integrate and noisy measurements to get robust estimates of the platform motion
  - A light touch of simultaneous localization and mapping (SLAM)
  - Project 3

Project 1

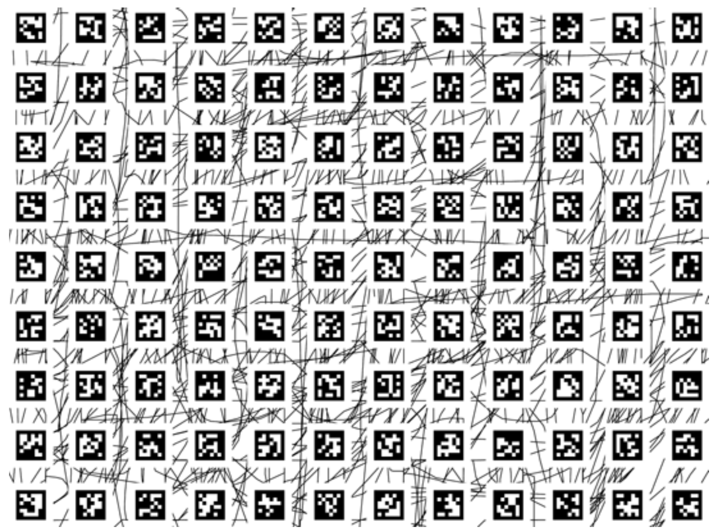
- Phase 1: Hover control of a simulated quadrotor
  - MATLAB, offline
- Phase 2: Trajectory generation and control of a simulated quadrotor
  - MATLAB, offline
- Phase 3: Trajectory control of a real quadrotor given perfect state estimation
  - ROS & C++, Desktop PC, online
  - Group Project (2-3 students/group)



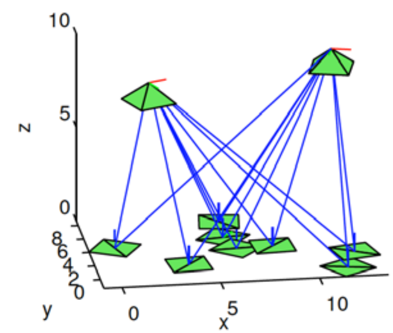


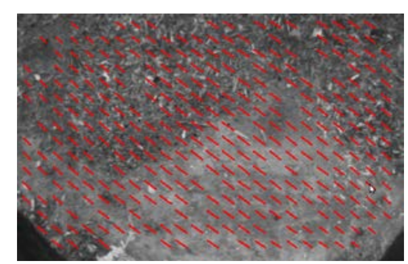
Project 2

- Phase 1: Vision-based position estimation
  - ROS & C++, Desktop PC, offline



- Phase 2: Vision-based velocity estimation
  - ROS & C++, Desktop PC, offline



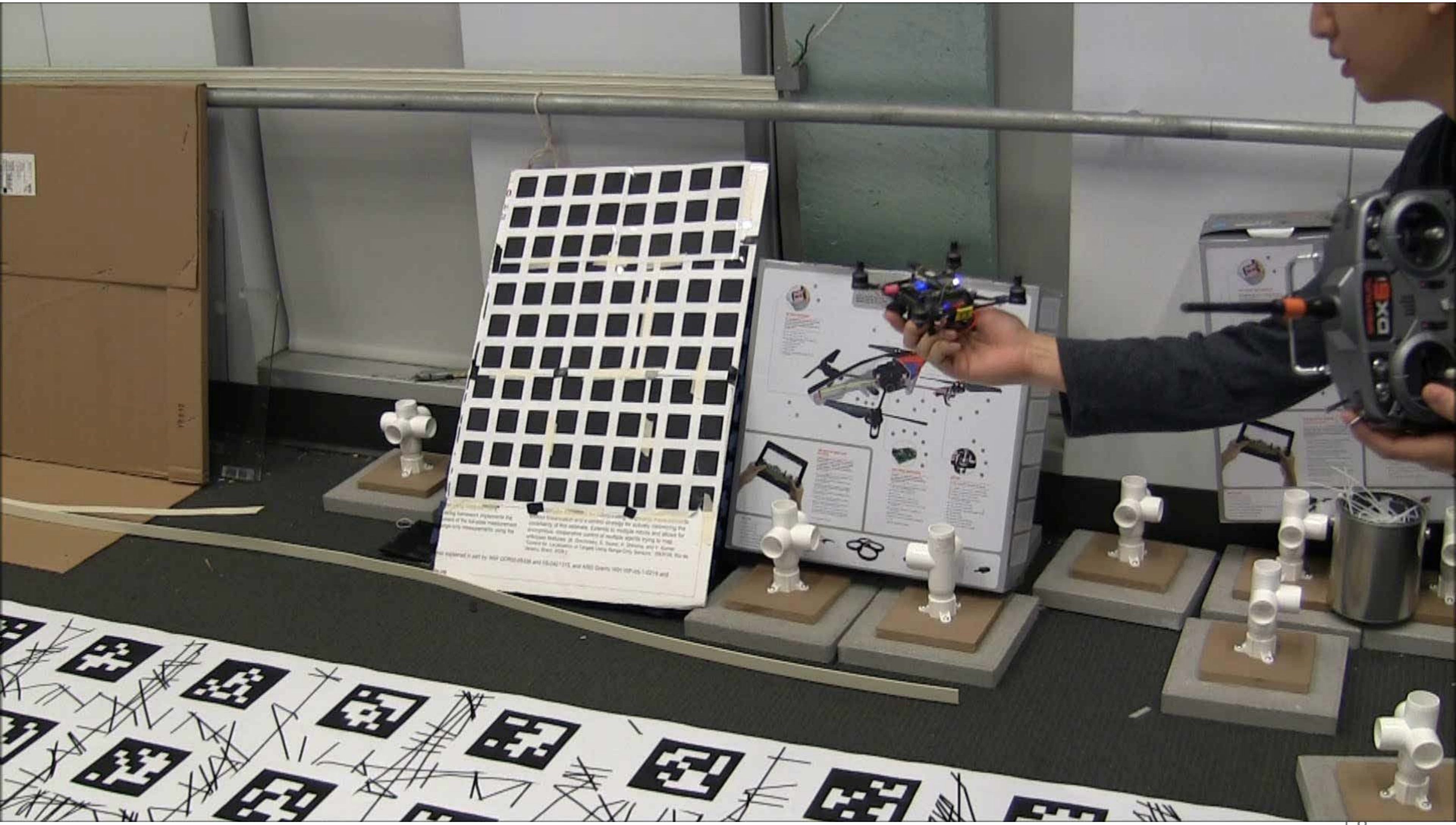


Project 3

- Phase 1: Extended Kalman filtering-based sensor fusion
  - ROS & C++, Desktop PC, offline

- Phase 2: Vision-based autonomous navigation
  - ROS & C++, Desktop PC & onboard, online
  - Group project (2-3 students/group)

Project 3



Rigid Body Transformations

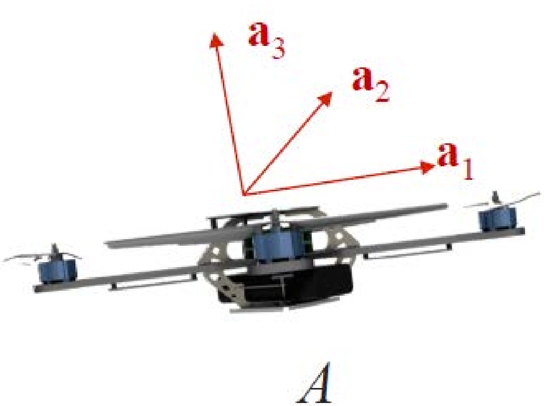
Rigid Body Transformations

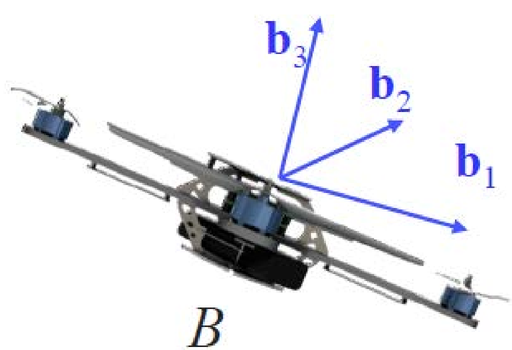
- Two distinct positions and orientations of the same rigid body

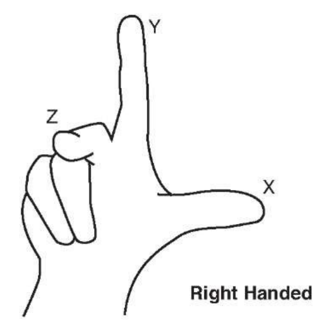
Courtesy Vijay Kumar

Reference Frames

- We associate any position and orientation with a reference frame
  - We use right-handed coordinate frames
  - We can find three linearly independent vectors $\mathbf{a}_1, \mathbf{a}_3, \mathbf{a}_3$ that are basis vectors for reference frame A
  - We can write any vector as a linear combination of basis vectors in either frame $\mathbf{v} = v_1\mathbf{a}_1 + v_2\mathbf{a}_2 + v_3\mathbf{a}_3$







Notation

- Vectors
  - **x, y, a**, …
- Matrices
  - **A, B, C, …**
- Reference frames
  - *A, B, C, …*
  - *a, b, c, …*
- Transformations
  - ${}^{A}\mathbf{A}_{B}$, ${}^{A}\mathbf{R}_{B}$ …
  - $\mathbf{A}_{ab}$, $\mathbf{R}_{ab}$ …
  - $g_{ab}(.)$, $h_{ab}(.)$ …

Be Aware of Potential Confusion!!!

Courtesy Vijay Kumar

Rigid Body Displacement

- Object:

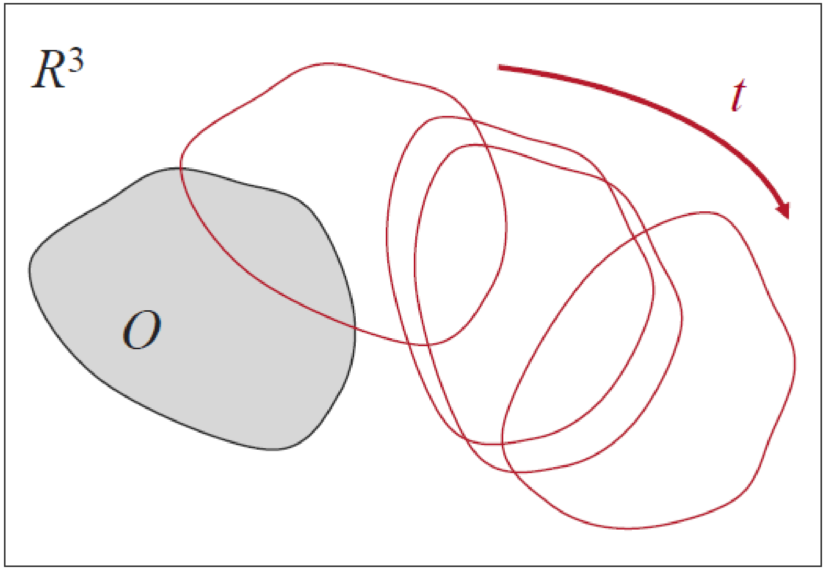
$$g \subset \Re^3$$

- Rigid body displacement
  - map

$$g : O \rightarrow \Re^3$$

- Rigid body motion
  - Continuous family of maps

$$g(t) : O \rightarrow \Re^3$$

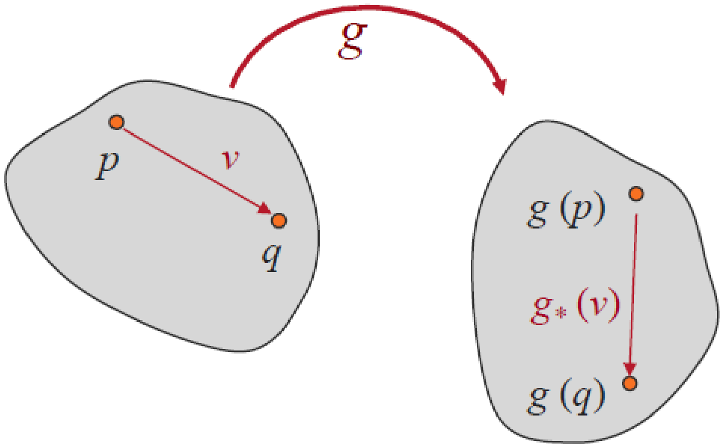


Courtesy Vijay Kumar

Rigid Body Displacement

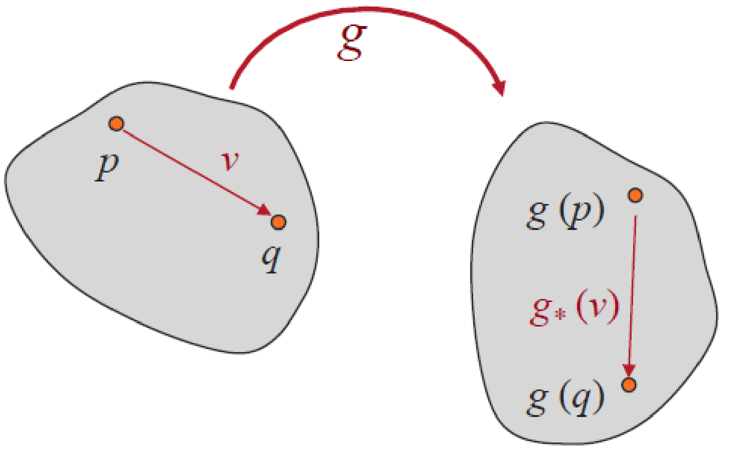
- A displacement of a transformation of points
  - Transformation (g) of points induces an action (g*) on vectors

$$g_*(\mathbf{v}) = g(\mathbf{q}) - g(\mathbf{p})$$



Courtesy Vijay Kumar

Definition of Rigid Body Displacement



- Lengths are preserved

$$\|g(\mathbf{q})-g(\mathbf{p})\|=\|\mathbf{q}-\mathbf{p}\|$$

- Cross products are preserved

$$g_*(\mathbf{v})\times g_*(\mathbf{w})=g_*(\mathbf{v}\times\mathbf{w})$$

Why?

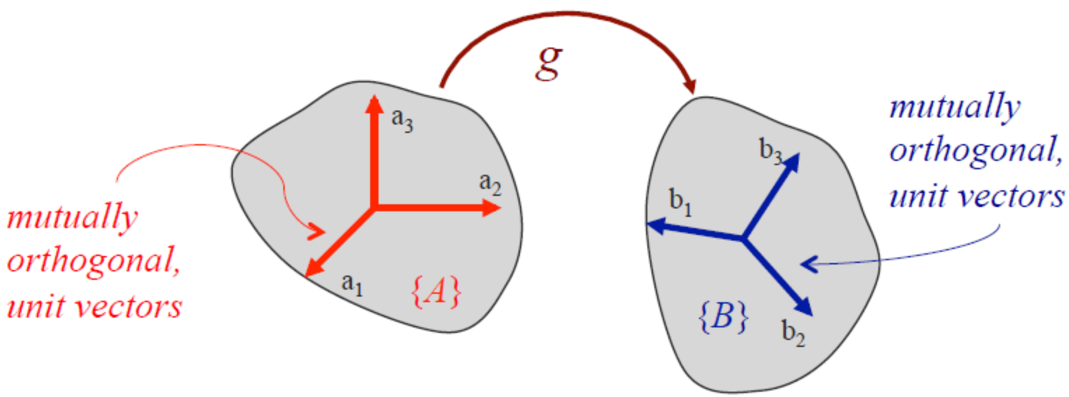
Eliminate internal reflection: $(x, y, z)\rightarrow(x, y, -z)$

Properties of Rigid Body Displacement

- Inner products are also preserved

$$g_*(\mathbf{v}) \cdot g_*(\mathbf{w}) = g_*(\mathbf{v} \cdot \mathbf{w})$$

- Orthogonal vectors are mapped to orthogonal vectors



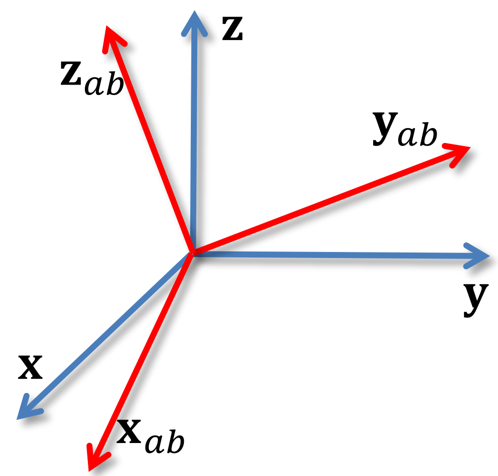
Rigid Body Displacement

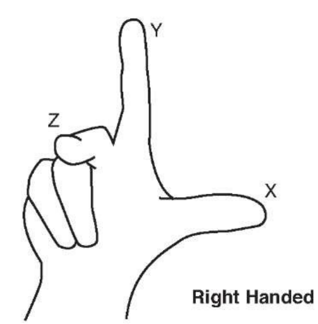
- Rigid body displacements are transformations that satisfy two important properties:
  1. Lengths are preserved
  2. Cross products are preserved

- Rigid body transformations and rigid body displacements are often used interchangeably
  – Transformations generally used to describe relationship between reference frames attached to different rigid bodies.
  – Displacements describe relationships between two positions and orientation of a frame attached to a displaced rigid body

Rotation

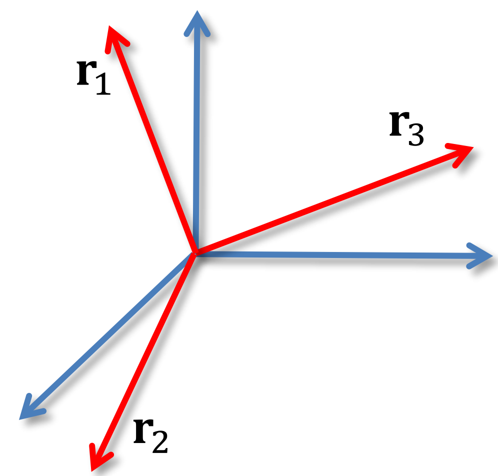
- Coordinate frames are right-handed
- Principle axes of frame *A*:
  - $\mathbf{x} = [1\ 0\ 0]^T$
  - $\mathbf{y} = [0\ 1\ 0]^T$
  - $\mathbf{z} = [0\ 0\ 1]^T$
- Principle axes of frame *B*:
  - $\mathbf{x}_{ab}, \mathbf{y}_{ab}, \mathbf{z}_{ab} \subset \mathbb{R}^3$
- The Rotation Matrix:
  - $\mathbf{R}_{ab} = [\mathbf{x}_{ab}, \mathbf{y}_{ab}, \mathbf{z}_{ab}]$
  - Coordinates of principle axes of *B* related to *A*





Properties of a Rotation Matrix

- Let $\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3]$ be a rotation matrix
- Orthogonal:
  - $\mathbf{r}_i{}^T \cdot \mathbf{r}_j = \begin{cases} 0 \text{ if } i \neq j \\ 1 \text{ if } i = j \end{cases}$
  - $\mathbf{R} \cdot \mathbf{R}^T = \mathbf{I}$
- Special orthogonal:
  - $\det\mathbf{R} = \mathbf{r}_1{}^T \cdot (\mathbf{r}_2 \times \mathbf{r}_3) = \mathbf{r}_1{}^T \cdot \mathbf{r}_1 = 1$



- The set of all rotations forms the Special Orthogonal Group
  - Special orthogonal group
  - 3D rotations: $SO(3)$
  - 2D rotations: $SO(2)$
  - $SO(n) = \{\mathbf{R} \in \mathbb{R}^{n \times n} | \mathbf{R} \cdot \mathbf{R}^T = \mathbf{I}, \det\mathbf{R} = 1\}$

Properties of a Rotation Matrix

- $SO(3) = \{\mathbf{R} \in \mathbb{R}^{3\times3} | \mathbf{R} \cdot \mathbf{R}^T = \mathbf{I}, \det\mathbf{R} = 1\}$

- $SO(3)$ is a group under the operation of matrix multiplication
  1. Closure: If $\mathbf{R}_1, \mathbf{R}_2 \in SO(3)$, then $\mathbf{R}_1 \cdot \mathbf{R}_2 \in SO(3)$
  2. Identity: The identity matrix is the identity element
  3. Inverse: If $\mathbf{R} \in SO(3)$, then $\mathbf{R}^{-1} \in SO(3)$
  4. Associativity: $\mathbf{R}_1 \cdot (\mathbf{R}_2 \cdot \mathbf{R}_3) = (\mathbf{R}_1 \cdot \mathbf{R}_2) \cdot \mathbf{R}_3$

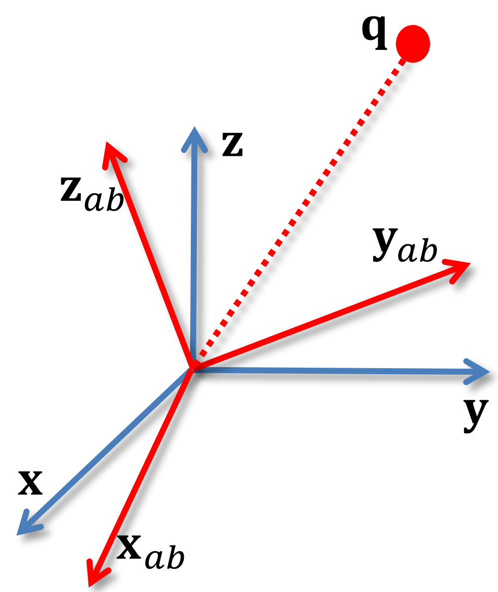
$(G, \cdot)$ is a group if:

1. $g_1, g_2 \in G \Rightarrow g_1 \cdot g_2 \in G$
2. $\exists!\ e \in G$, s.t. $g \cdot e = e \cdot g = g, \forall g \in G$
3. $\forall g \in G, \exists!\ g^{-1} \in G$, s.t. $g \cdot g^{-1} = g^{-1} \cdot g = e$
4. $g_1 \cdot (g_2 \cdot g_3) = (g_1 \cdot g_2) \cdot g_3$

Courtesy Zexiang Li

Properties of a Rotation Matrix

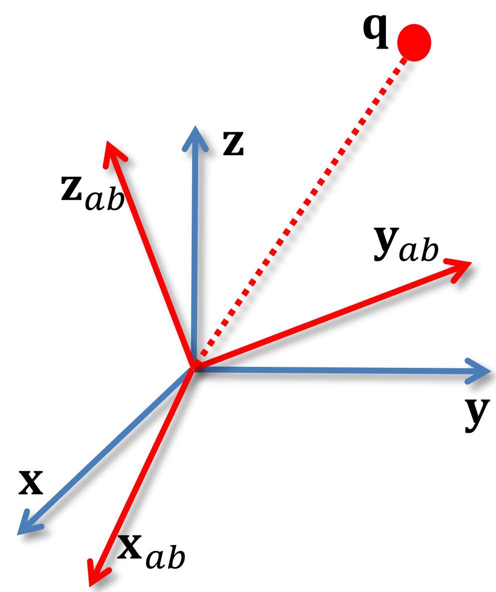
- A transformation that rotates the coordinates of a point from frame *B* to frame *A*
  - Let $\mathbf{q}_b = [x_b, y_b, z_b]^T \in \mathbb{R}^3$ be coordinate of point $\mathbf{q}$ in frame *B*
  - Let $\mathbf{q}_a$ be coordinate of point $\mathbf{q}$ in frame *A*
  - $\mathbf{q}_a = x_b \cdot \mathbf{x}_{ab} + y_b \cdot \mathbf{y}_{ab} + z_b \cdot \mathbf{z}_{ab} = [\mathbf{x}_{ab}, \mathbf{y}_{ab}, \mathbf{z}_{ab}] \begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} = \mathbf{R}_{ab} \cdot \mathbf{q}_b$



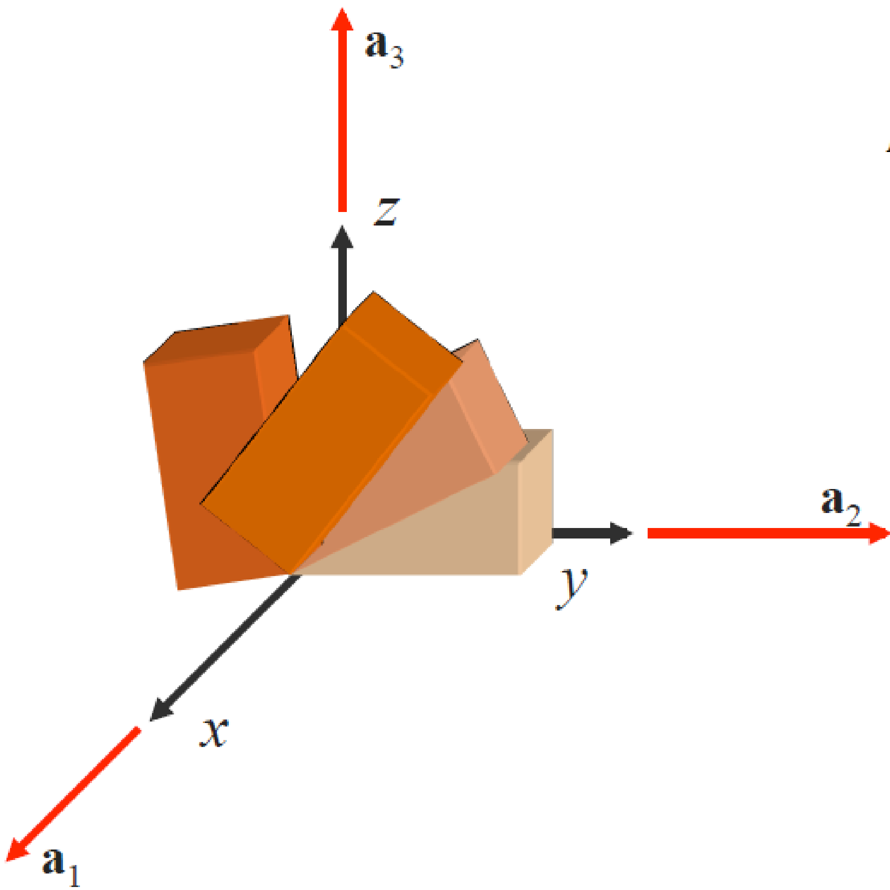
- Composition Rule
  - $\mathbf{R}_{ac} = \mathbf{R}_{ab} \cdot \mathbf{R}_{bc}$

Rotation is Rigid Body Transformation

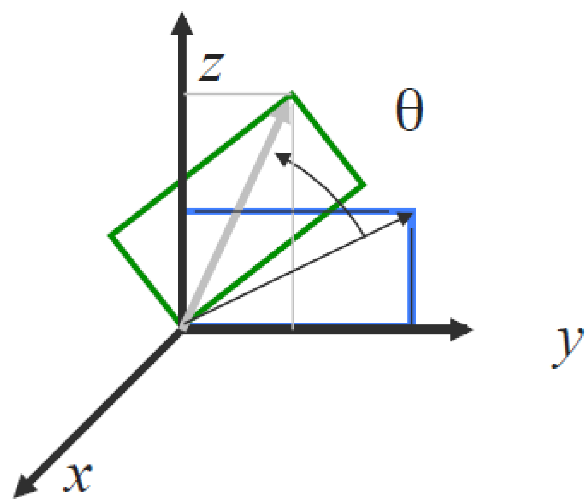
- $\mathbf{R}_{ab} = [\mathbf{x}_{ab}, \mathbf{y}_{ab}, \mathbf{z}_{ab}]$ preserves:
  - Length:
    - $\|\mathbf{R}_{ab}(\mathbf{p}_b - \mathbf{q}_b)\| = \|\mathbf{p}_b - \mathbf{q}_b\|$
  - Cross product:
    - $\mathbf{R}_{ab}(\mathbf{v} \times \boldsymbol{w}) = (\mathbf{R}_{ab}\mathbf{v}) \times (\mathbf{R}_{ab}\mathbf{w})$
    - Use the fact $\mathbf{R}(\mathbf{v})^{\wedge}\mathbf{R}^T = (\mathbf{R}\mathbf{v})^{\wedge}$ to prove, where $(\mathbf{a})^{\wedge} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$ is the skew-symmetric matrix, and $\mathbf{a} \times \mathbf{b} = (\mathbf{a})^{\wedge}\mathbf{b}$



Example - Rotation



$$Rot(x,\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$$

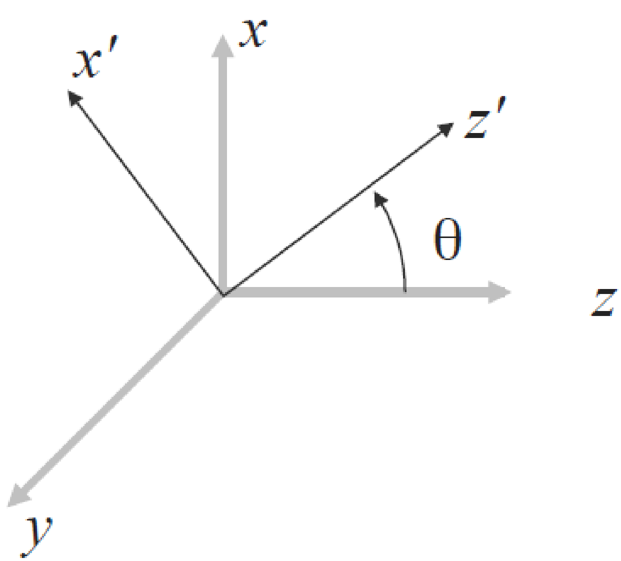


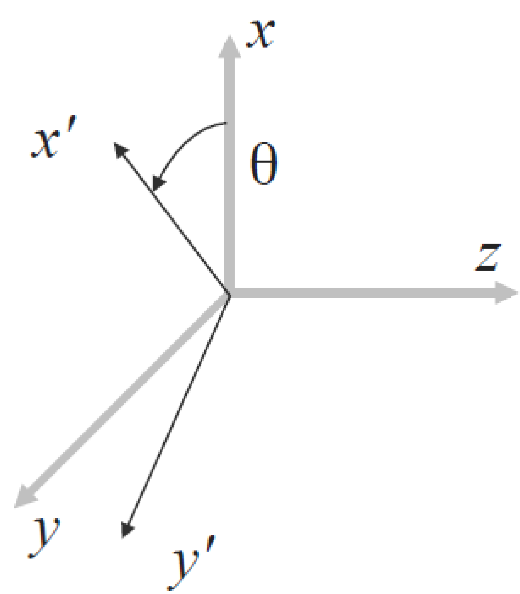
Courtesy Vijay Kumar

Example - Rotation

$$Rot(y,\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

$$Rot(z,\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$





Courtesy Vijay Kumar

Assignment

- Form a project group
- Review C++ if you are not very good at it
- Chapter 2.1-2.3 of "A Mathematical Introduction to Robotic Manipulation"

Next Lecture…

- Rotation representations
- Rigid body motions
- Quadrotor modeling