UNIVERSITY OF SOUTH CAROLINA

CSCE 774 ROBOTIC SYSTEMS

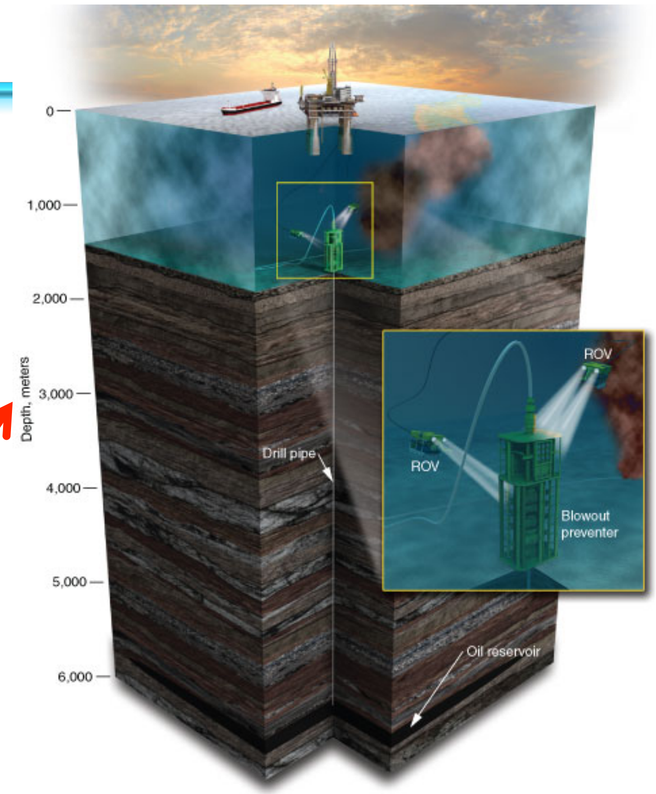
Introduction

Fall 2015

Ioannis Rekleitis

Present Everywhere

- At home
- On the road
- In the sky (drones)
- In the fields (agricultural robotics)
- In resource utilization (**ROV in the oil industry**)
- Along power lines
- In Hospitals
- Education



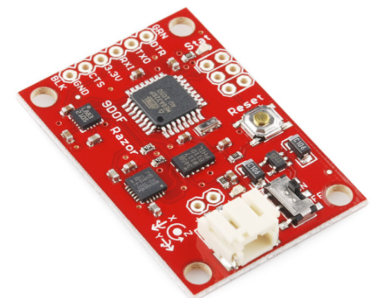
Robotic technology becomes affordable

TurtleBot 2

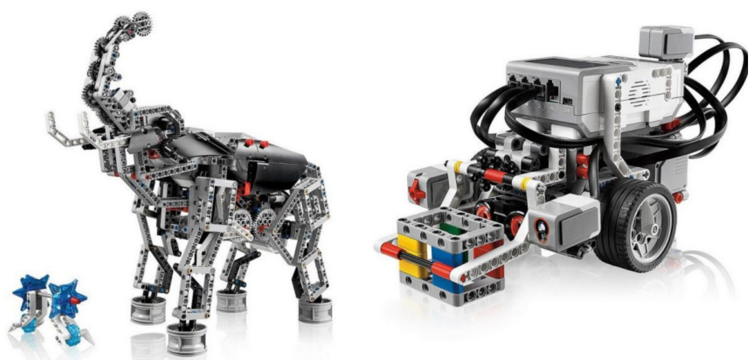
AR.DRONE

Kinect

IMU

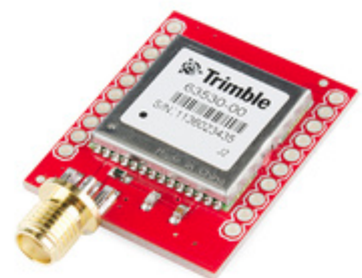


Raspberry Pi

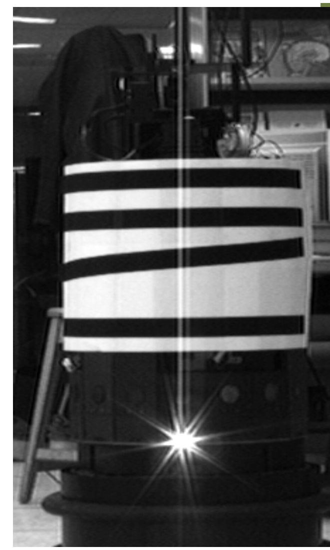


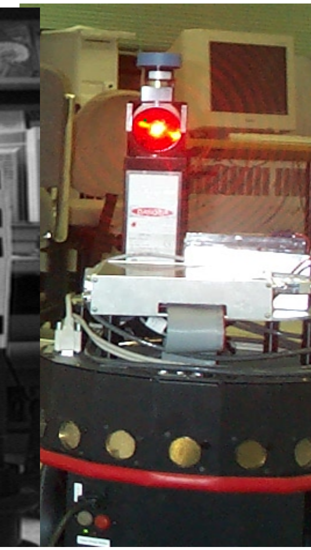
Lego Mindstorm

GPS

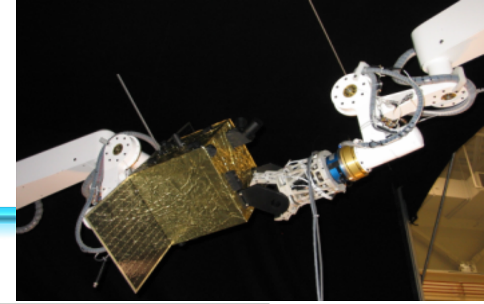


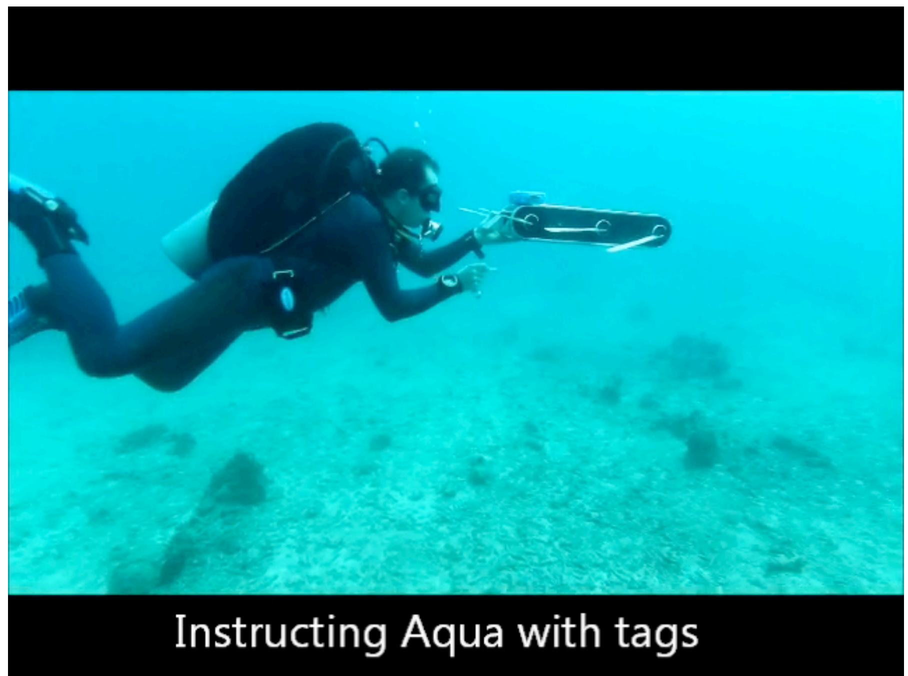
Past Projects





Past/Current Projects





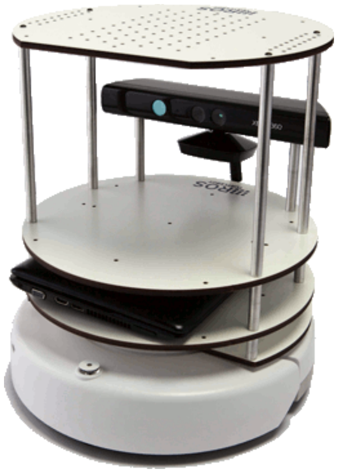
Current work in U/W Robotics

Center for Computational Robotics (CCR)

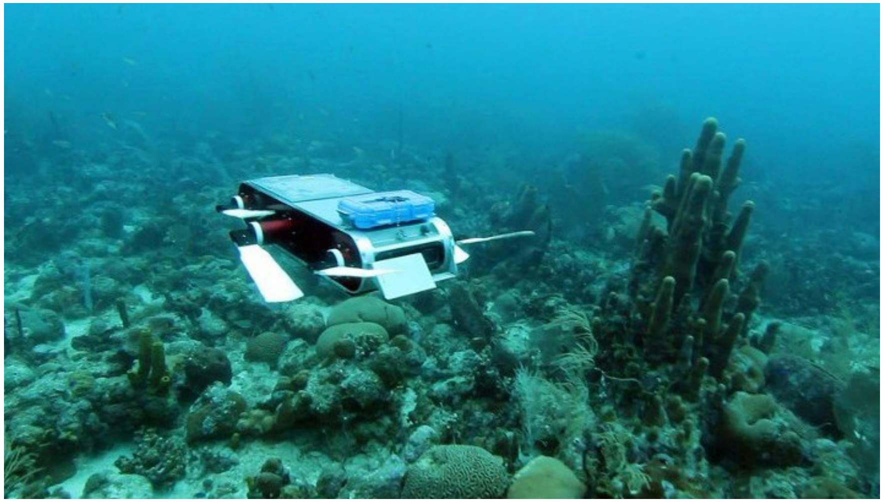
- SCARR lab – Jason O'Kane
- ART lab – Jenay Beer
- AFRL – Ioannis Rekleitis

- AFRL: Autonomous Field Robotics Lab

Existing CCR Robotic Platforms



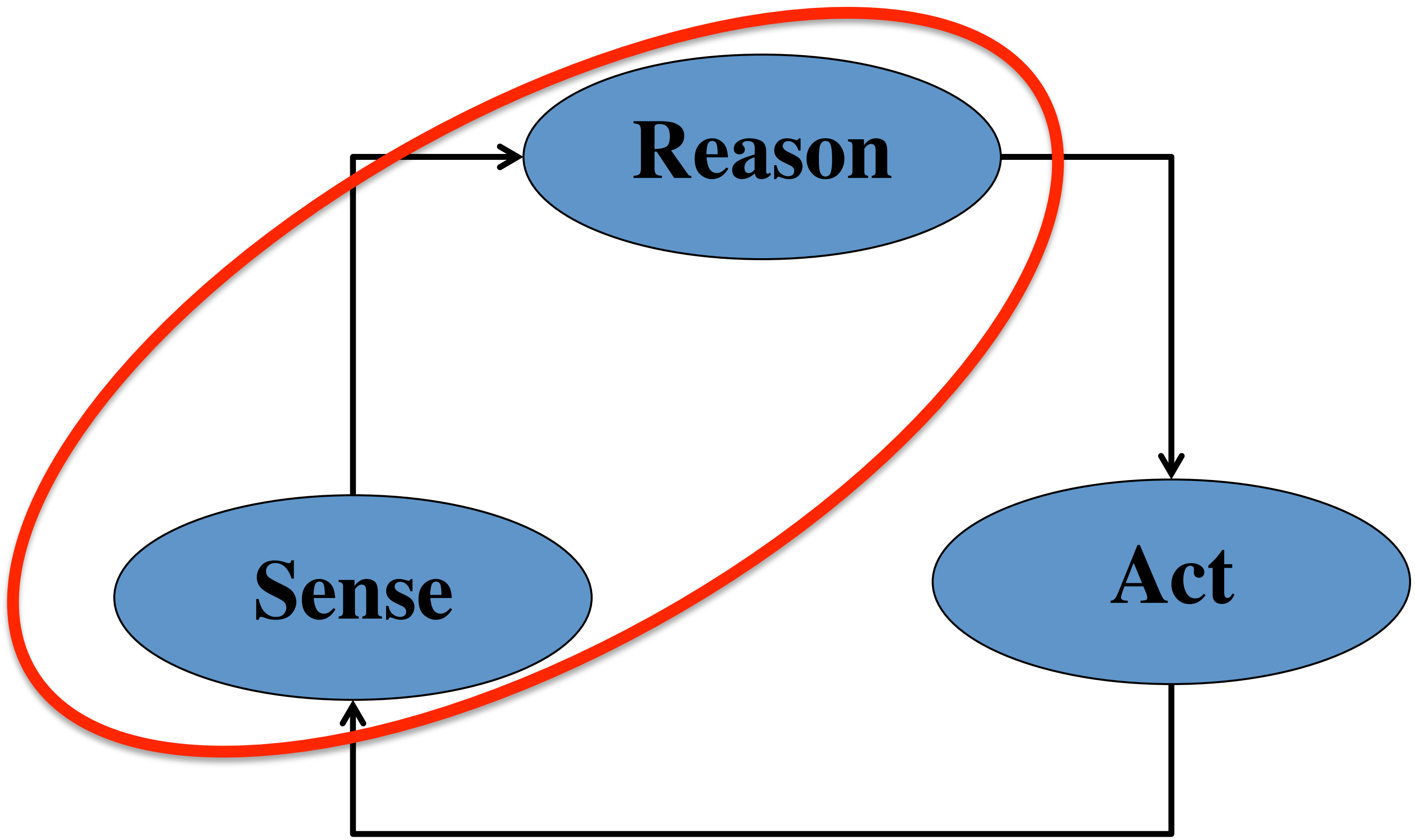
Upcoming CCR Robotic Platforms



Three Main Challenges in Robotics

1. Where am I? (Localization)
2. What the world looks like? (Mapping)
   – Together 1 and 2 form the problem of *Simultaneous Localization and Mapping* (SLAM)
3. How do I go from **A** to **B**? (Path Planning)
   – More general: Which action should I pick next? (Planning)

Robot



Syllabus

- Focus on **Localization**, **Mapping**, and **SLAM**
- Reading and discussing different research papers
- Presentations by students
- Hands on assignments

Evaluation

• 3 Homeworks, 10% each:	30%
• Final Examination:	20%
• Class Participation:	20%
• Presentations:	30%

Homeworks/Projects

- Using ROS and OpenCV
- Using Simulations
- Using sensor data from real robots
- Using real robots (TurtleBot)

Papers

1. H. W. Sorenson. Least-squares estimation from Gauss to Kalman, 1970
2. H. Durrant-Whyte and T. Bailey. Simultaneous Localisation and Mapping: Part I, 2006
3. T. Bailey and H. Durrant-Whyte. Simultaneous Localisation and Mapping: Part II, 2006
4. R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotic", 1990
5. S. J. Julier J. K. Uhlmann. A New Extension of the Kalman Filter to Nonlinear Systems
6. F. Lu and E. Milios, Globally consistent range scan alignment for environment mapping
7. F. Lu and E. Milios, Robot pose estimation in unknown environments by matching 2d range scans
8. G. Grisetti, C. Stachniss, and W. Burgard. Improved Techniques for Grid Mapping with Rao-Blackwellized Particle Filters
9. D. Scaramuzza, F. Fraundorfer. Visual Odometry: Part I - The First 30 Years and Fundamentals 2011.
10. F. Fraundorfer, D. Scaramuzza. Visual odometry: Part II - Matching, robustness, optimization, and applications. 2012.
11. D. Nister O. Nardoditsky, and J. Bergen. Visual odometry for ground vehicle applications, 2006
12. B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle Adjustment — A Modern Synthesis
13. G. Klein and D. Murray. Parallel Tracking and Mapping for Small AR Workspaces, 2007
14. A.I. Mourikis and S.I. Roumeliotis. A Multi-State Constrained Kalman filter for Vision-aided Inertial Navigation, 2007
15. E. Jones and S. Soatto. Visual-Inertial Navigation, Mapping and Localization: A Scalable Real-Time Causal Approach, 2011.
16. R. Mur-Artal, J. M. M. Montiel and J. D. Tardós. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. 2015
17. M. Cummins and P. Newman. FAB-MAP: Probabilistic localization and mapping in the space of appearance, 2008
18. G. Sibley C. Mei, I. Reid, and P. Newman. Adaptive relative bundle adjustment, 2009
19. M. Milford, G. Wyeth. Persistent navigation and mapping using a biologically inspired SLAM system, 2010
20. C. Forster, M. Pizzoli, and D. Scaramuzza. SVO: Fast Semi-Direct Monocular Visual Odometry, 2014
21. J. Engel, T. Schöps, D. Cremers. LSD-SLAM: Large-Scale Direct Monocular SLAM, 2014.
22. R. A. Newcombe S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, A. Fitzgibbon. KinectFusion: Real-Time Dense Surface Mapping and Tracking, 2011
23. S. Thrun and M. Montemerlo. The GraphSLAM Algorithm with Applications to Large-Scale Mapping of Urban Structures, 2006
24. I. Mahon, O. Pizarro, M. Johnson-Roberson, A. Friedman, S. Williams, J. Henderson. Reconstructing Pavlopetri: mapping the world's oldest submerged town using stereo-vision, 2011
25. F. Shkurti, I. Rekleitis, M. Scaccia, G. Dudek. State estimation of an underwater robot using visual and inertial information, 2011
26. R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. G2O: A general framework for graph optimisation, 2011
27. M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Incremental Smoothing and Mapping, 2008
28. M. Kaess, H. Johannsson, R. Roberts, V. Ila, J.J. Leonard, and F. Dellaert. iSAM2: Incremental Smoothing and Mapping Using the Bayes Tree. 2012
29. M. Montemerlo, S. Thrun, D. Koller, and B.Wegbreit. FastSLAM 2.0: An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping that Provably Converges, 2003

Contact

- http://www.cse.sc.edu/~yiannisr/
- http://www.cse.sc.edu/~yiannisr/774/2015

- **Email**: yiannisr@cse.sc.edu

- **Office hours**: 3A54 -- Tue/Th 13:00-14:00
  and by appointment

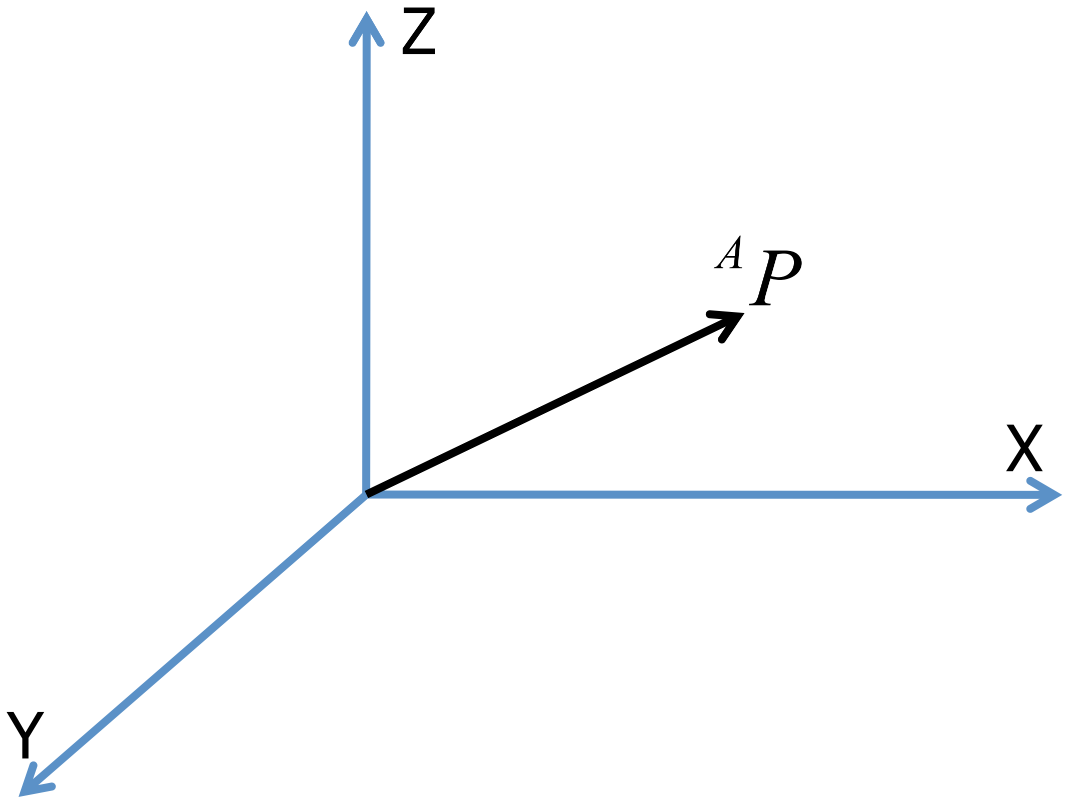
Class Interests

- Introductions
- Background
- Interests
- Projects
- Reasons
- Expectations

Position Representation

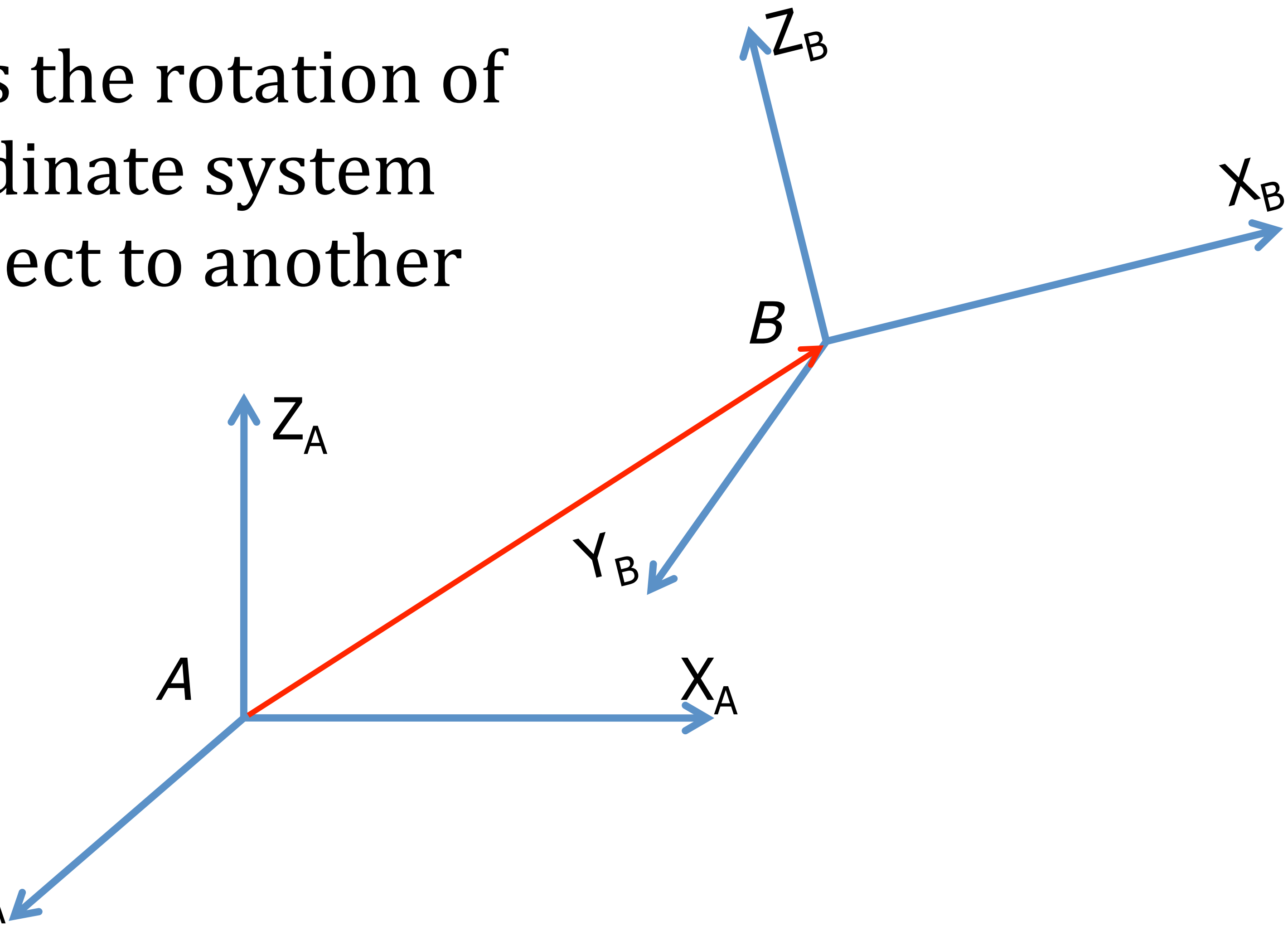
- Position representation is:

$$ {}^{A}P = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} $$



Orientation Representations

- Describes the rotation of one coordinate system with respect to another

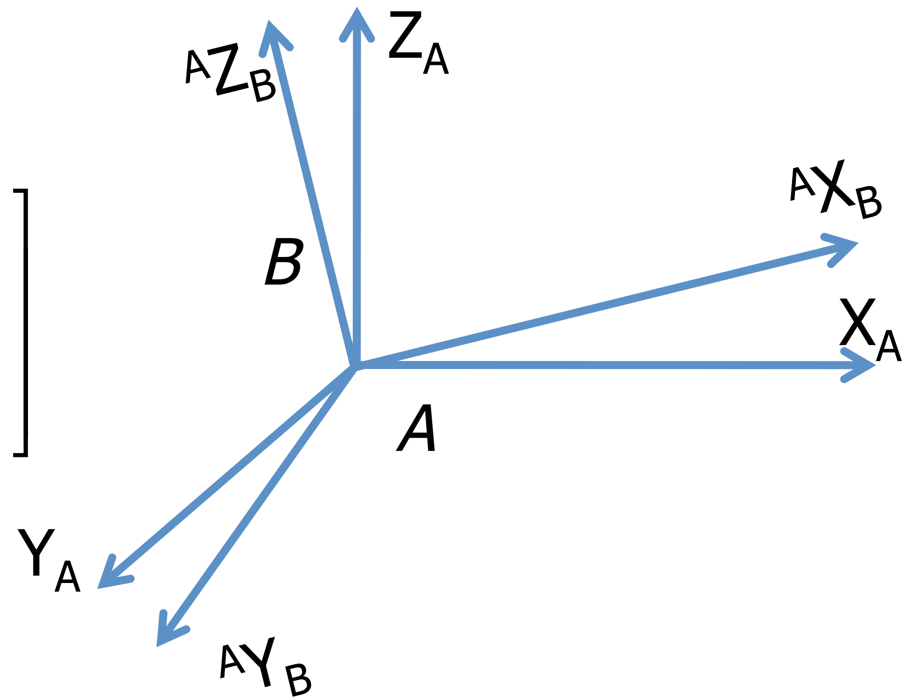


Rotation Matrix

- Write the unit vectors of B in the coordinate system of A.
- Rotation Matrix:

$$
{}^{A}_{B}R = \begin{bmatrix} {}^{A}\hat{X}_{B} & {}^{A}\hat{Y}_{B} & {}^{A}\hat{Z}_{B} \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}
$$

$$
= \begin{bmatrix} \hat{X}_{B} \cdot \hat{X}_{A} & \hat{Y}_{B} \cdot \hat{X}_{A} & \hat{Z}_{B} \cdot \hat{X}_{A} \\ \hat{X}_{B} \cdot \hat{Y}_{A} & \hat{Y}_{B} \cdot \hat{Y}_{A} & \hat{Z}_{B} \cdot \hat{Y}_{A} \\ \hat{X}_{B} \cdot \hat{Z}_{A} & \hat{Y}_{B} \cdot \hat{Z}_{A} & \hat{Z}_{B} \cdot \hat{Z}_{A} \end{bmatrix}
$$



Coordinate System Transformation

$$M = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R & T \\ 0_{3\times 1} & 1 \end{bmatrix}$$

where R is the rotation matrix and T is the translation vector

Rotation Matrix

- The rotation matrix consists of 9 variables, but there are many constraints. The minimum number of variables needed to describe a rotation is three.

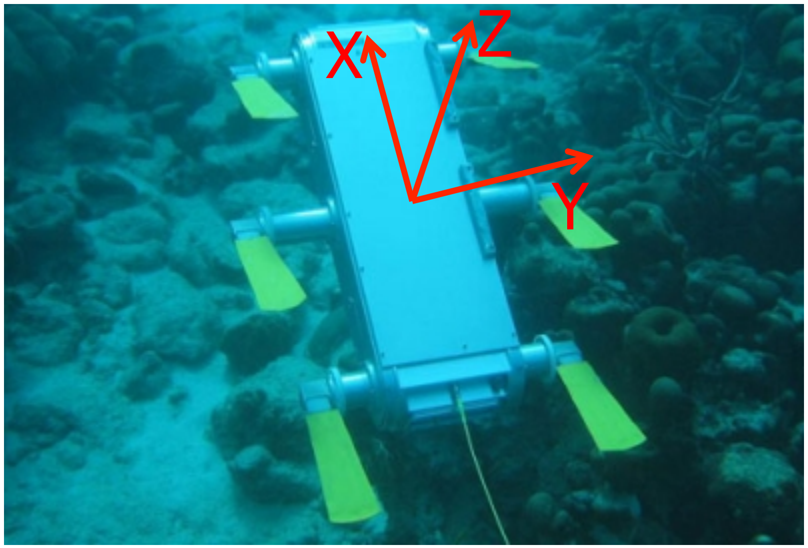
Euler Angles

- **ZYX**: Starting with the two frames aligned, first rotate about the Z_B axis, then by the Y_B axis and then by the X_B axis. The results are the same as with using XYZ fixed angle rotation.

- There are 12 different combination of Euler Angle representations

Euler Angles

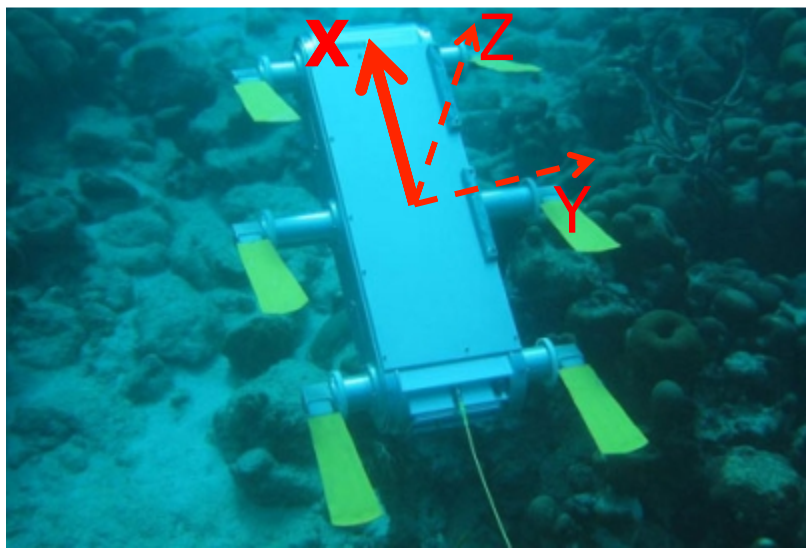
- Traditionally the three angles along the axis are called Roll, Pitch, and Yaw



Euler Angles

- Traditionally the three angles along the axis are called Roll, Pitch, and Yaw

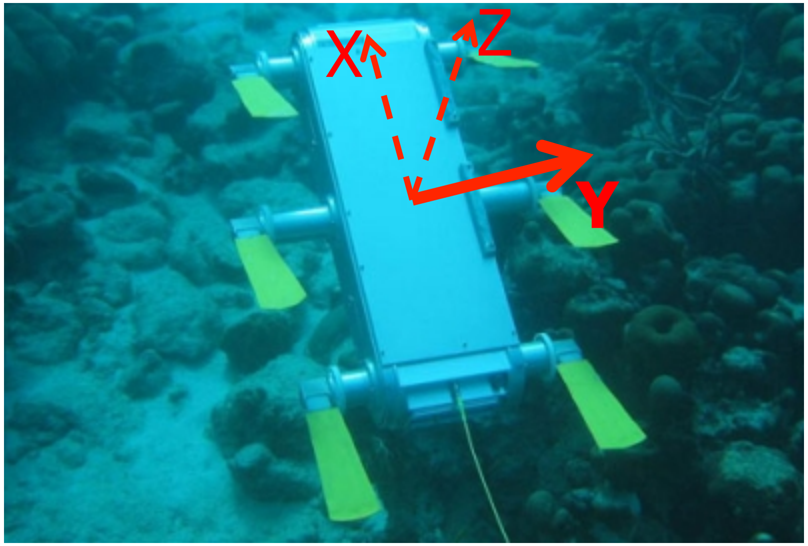
Roll



Euler Angles

- Traditionally the three angles along the axis are called Roll, Pitch, and Yaw

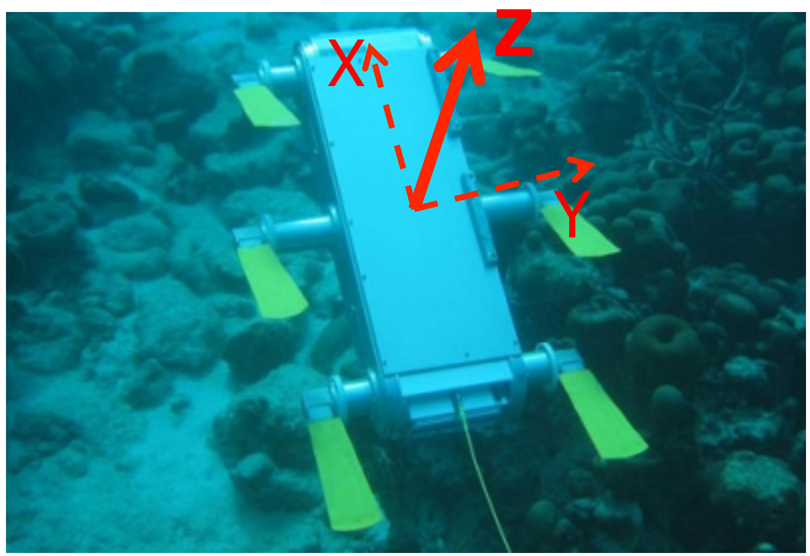
Pitch



Euler Angles

- Traditionally the three angles along the axis are called Roll, Pitch, and Yaw

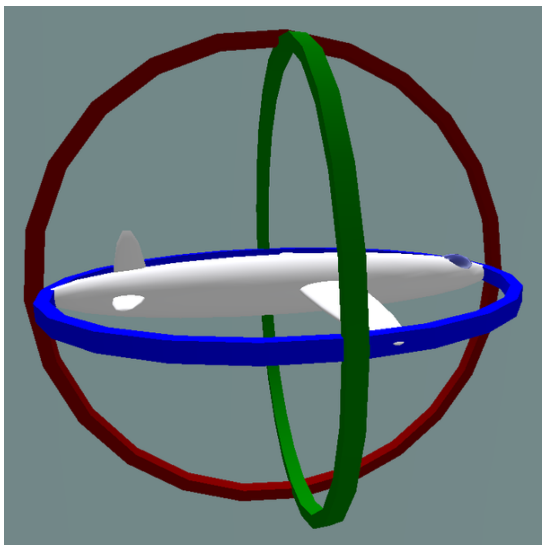
Yaw

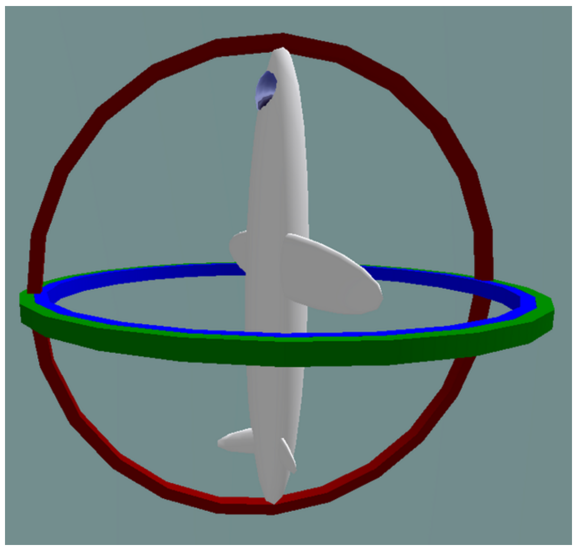


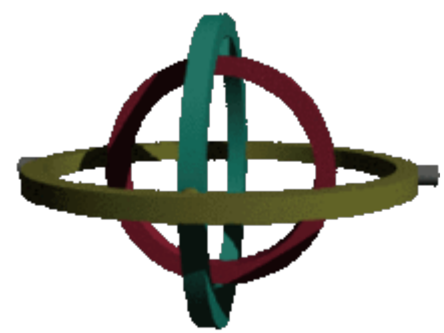
Euler Angle concerns: Gimbal Lock

Using the **ZYZ** convention

- (90°, 45°, −105°) ≡ (−270°, −315°, 255°) multiples of 360°
- (72°, 0°, 0°) ≡ (40°, 0°, 32°) singular alignment (Gimbal lock)
- (45°, 60°, −30°) ≡ (−135°, −60°, 150°) bistable flip

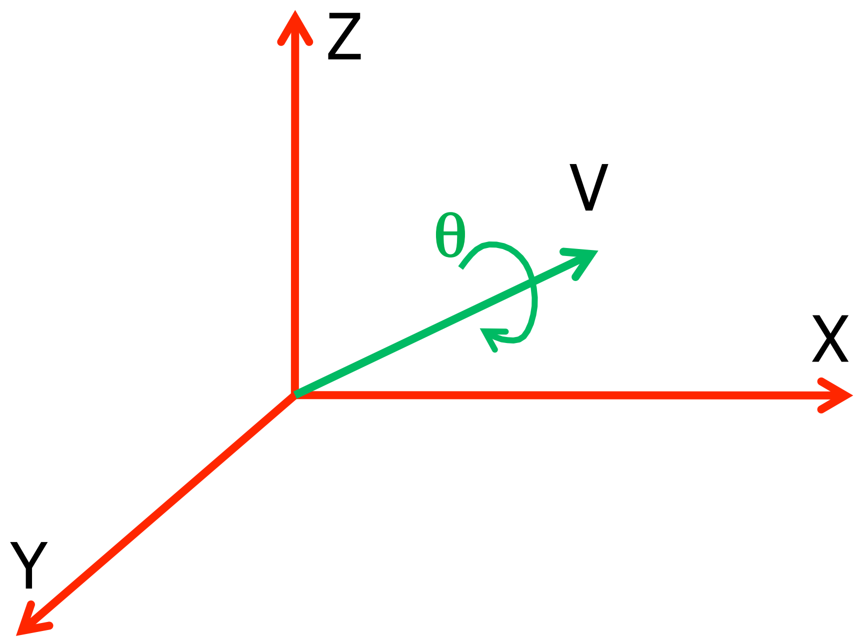






Axis-Angle Representation

- Represent an arbitrary rotation as a combination of a vector and an angle



Quaternions

- Are similar to axis-angle representation
- Two formulations
  - Classical
  - Based on JPL's standards

    W. G. Breckenridge, "Quaternions - Proposed Standard Conventions," JPL, Tech. Rep. INTEROFFICE MEMORANDUM IOM 343-79-1199, 1999.
- Avoids Gimbal lock

- See also: M. D. Shuster, "A survey of attitude representations," Journal of the Astronautical Sciences, vol. 41, no. 4, pp. 439–517, Oct.–Dec. 1993.

Robot Sensors

- Sensors are devices that can sense and measure physical properties of the environment,
  - e.g. temperature, luminance, resistance to touch, weight, size, etc.
  - The key phenomenon is transduction
    - Transduction (engineering) is a process that converts one type of energy to another
- They deliver *low-level* information about the environment the robot is working in.
  - Return an incomplete description of the world.

Robot Sensors

- This information is **noisy** (imprecise).
- Cannot be modelled completely:
  - Reading = *f*(env)   where *f* is the model of the sensor
  - Finding the inverse:
    - ill posed problem (solution not uniquely defined)
    - collapsing of dimensionality leads to ambiguity

Types of sensor

- General classification:
  - **active** *versus* **passive**
    - Active: emit energy in environment
      - More robust, less efficient
    - Passive: passively receive energy from env.
      - Less intrusive, but depends on env. e.g. light for camera
    - Example: stereo vision versus range finder.
  - **contact** *versus* **non-contact**

Sensors

- **Proprioceptive Sensors**
  (monitor state of robot)
  - IMU (accels & gyros)
  - Wheel encoders
  - Doppler radar …

- **Exteroceptive Sensors**
  (monitor environment)
  - Cameras (single, stereo, omni, FLIR …)
  - Laser scanner
  - MW radar
  - Sonar
  - Tactile…

Sensor Characteristics

- All sensors are characterized by various properties that describe their capabilities
  - **Sensitivity**:

    (change of output) ÷ (change of input)
  - **Linearity**: constancy of (output ÷ input)
    - Exception: logarithmic response cameras == wider dynamic range.
  - **Measurement/Dynamic range**:

    difference between min. and max.

Sensor Characteristics

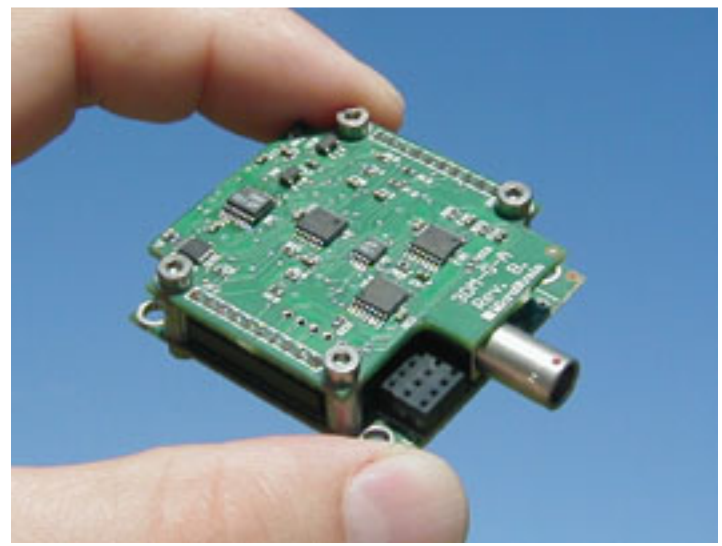
- **Response Time**: time required for a change in input to cause a change in the output
- **Accuracy**: difference between measured & actual
- **Repeatability**: difference between repeated measures
- **Resolution**: smallest observable increment
- **Bandwidth**: result of high resolution or cycle time

Focus on:

- IMU
- Wheel Encoders
- Compass
- Monocular Vision
- Stereo Vision
- RGBd (Kinnect)
- LIDAR

IMU's

- Gyro, accelerometer combination.
- Typical designs (e.g. 3DM-GX1™) use tri-axial gyros to track dynamic orientation and tri-axial DC accelerometers along with the tri-axial magnetometers to track static orientation.
- The embedded microprocessors contains programmable filter algorithms, which blend these static and dynamic responses in real-time.



Why vision?

- Passive (emits nothing).
  - Discreet.
  - Energy efficient.
- Intuitive.
- Powerful (works well for us, right?)
- Long and short range.
- Fast.

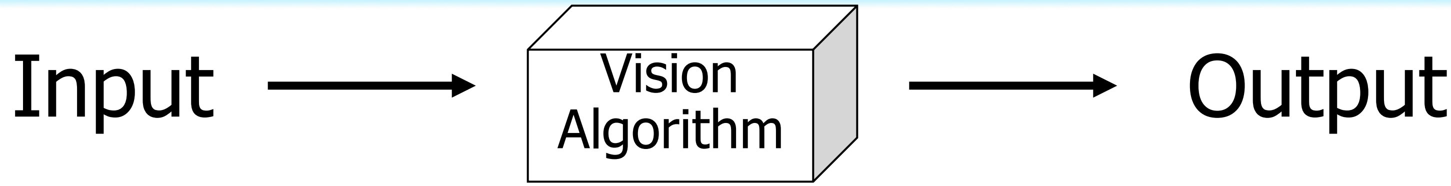
So, what's the problem?

- How hard is vision? Why do we think is do-able?

Problems:

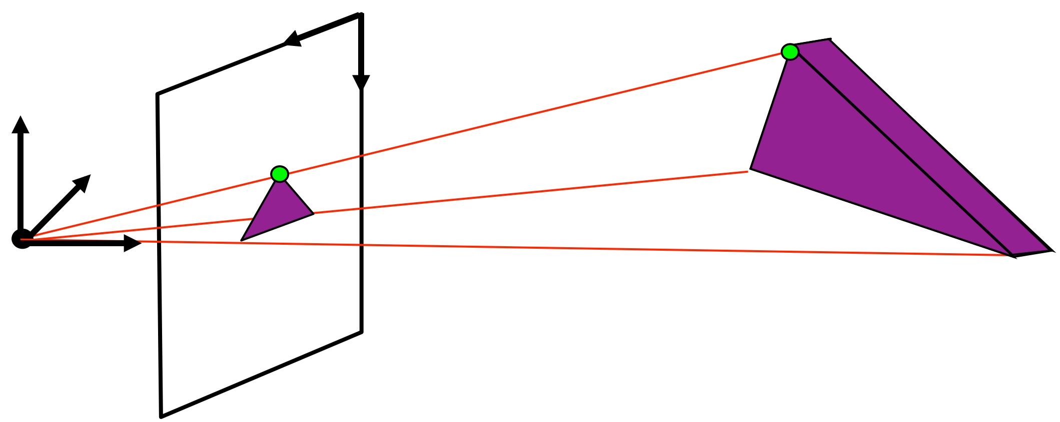
- Slow.
- Data-heavy.
- Impossible.
- Mixes up many factors.

The "Vision Problem"



The vision problem in general…

- In trying to extract 3d structure from 2d images, vision is an *ill-posed* problem.
- Basically, there are too many possible worlds that might (in theory) give rise to a particular image

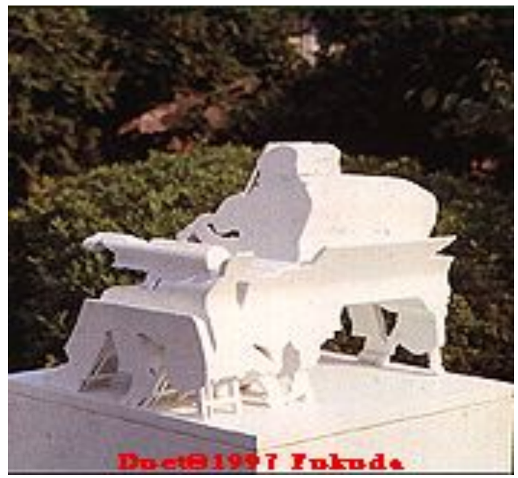


Ill-posed

- In trying to extract 3d structure from 2d images, vision is an *ill-posed* problem.

Ill-posed

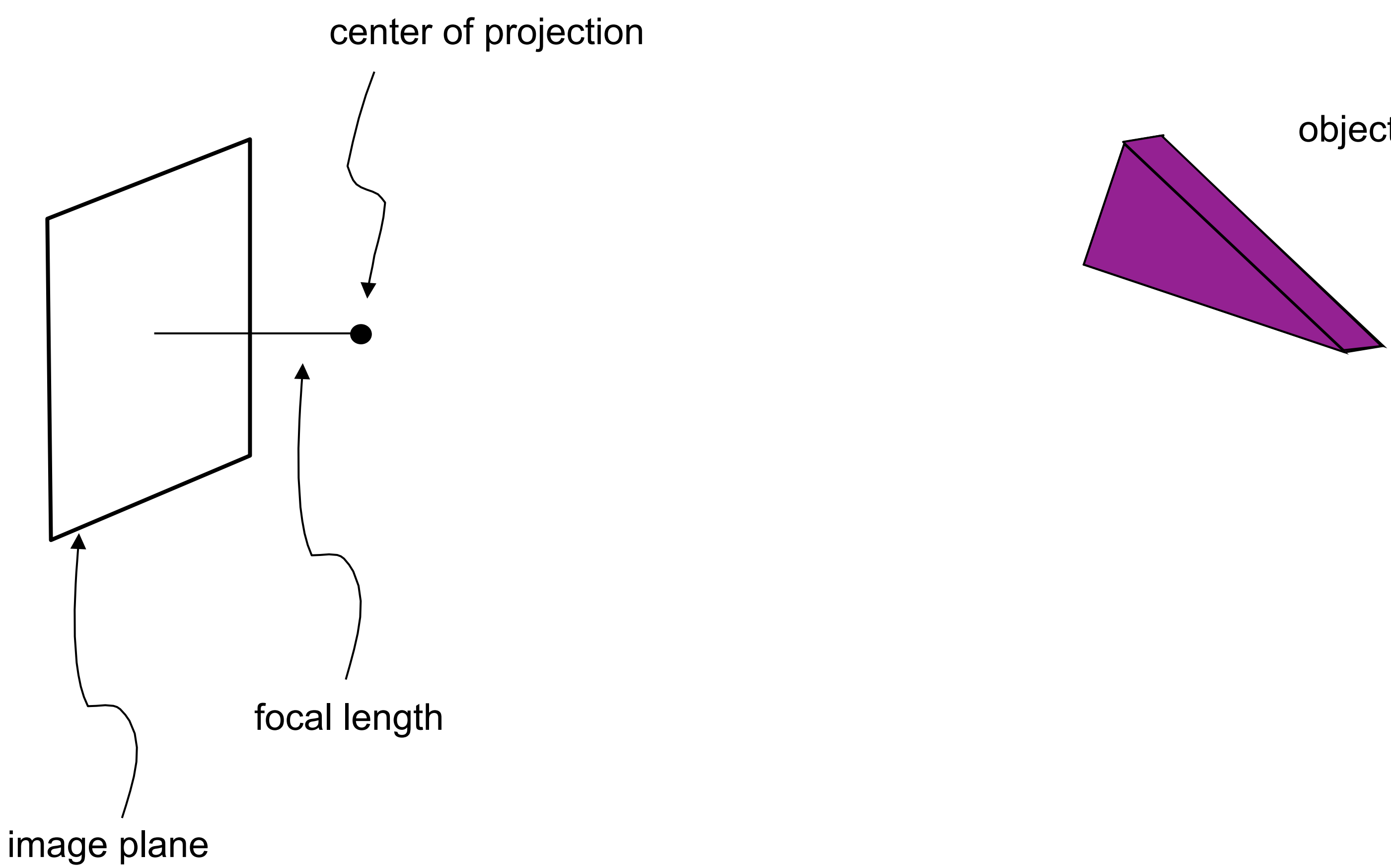
- In trying to extract 3d structure from 2d images, vision is an *ill-posed* problem.



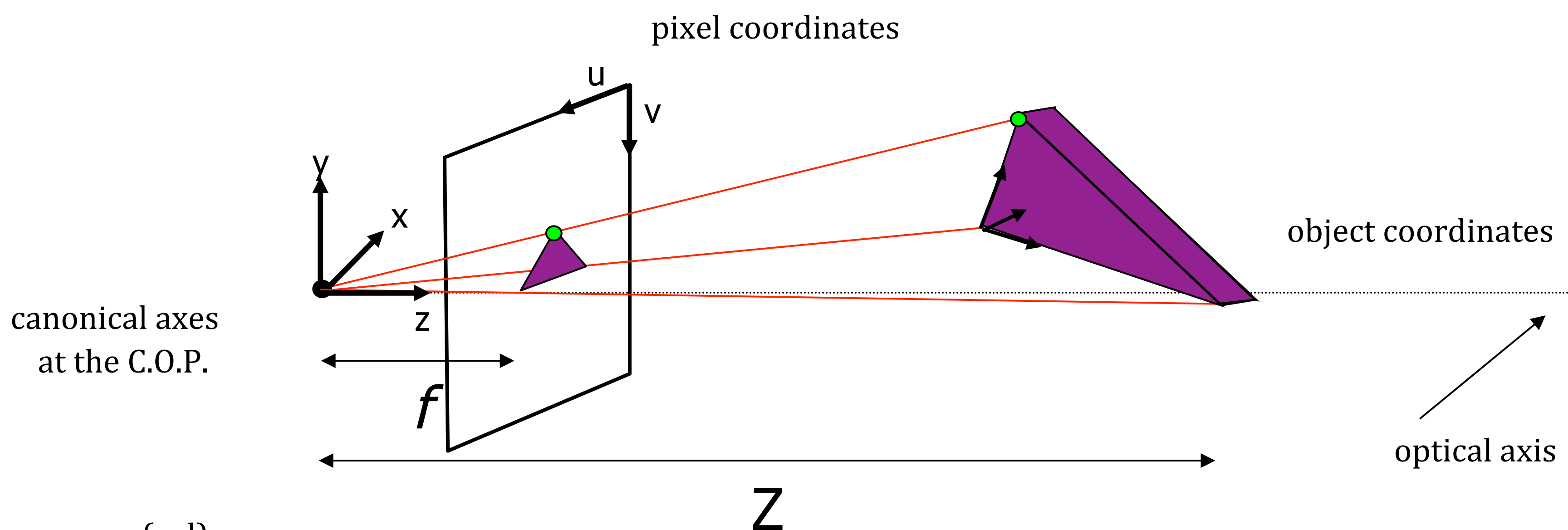
  – An image isn’t enough to disambiguate the many possible 3d worlds that could have produced it.

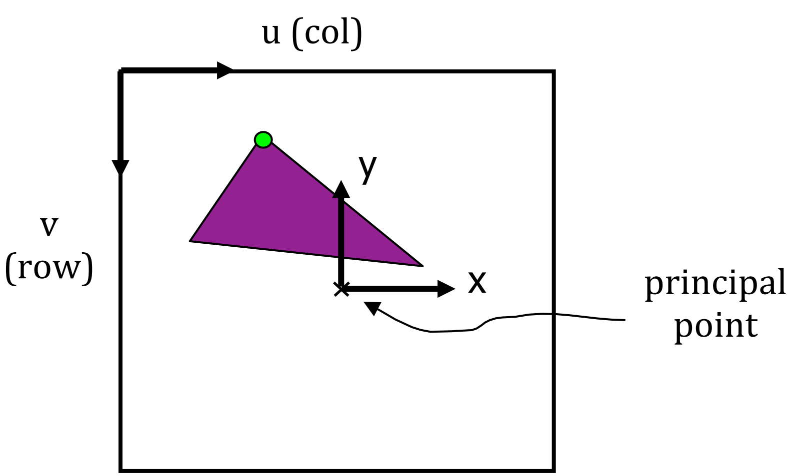
Camera Geometry

3D→2D transformation: perspective projection



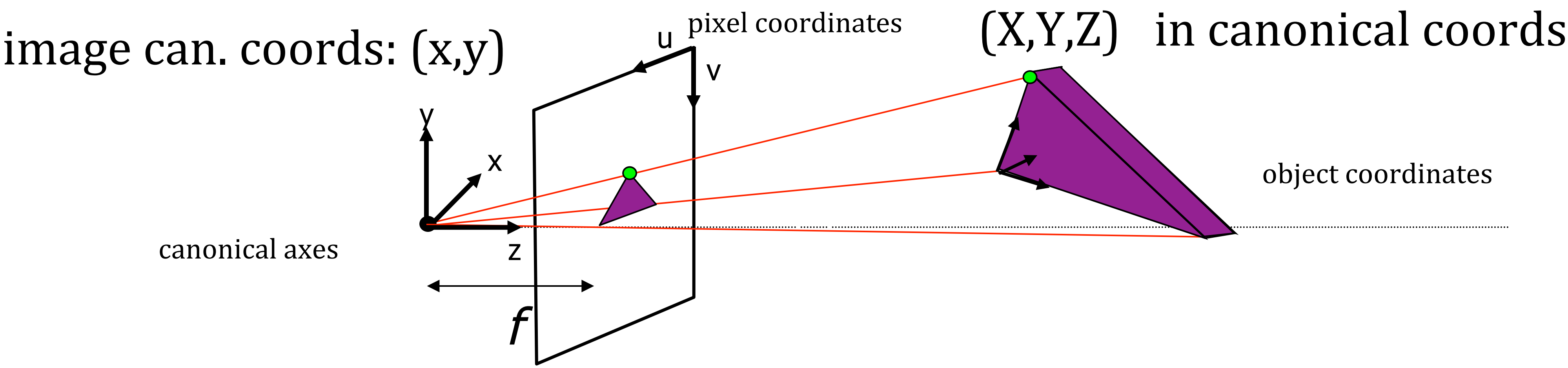
Coordinate Systems



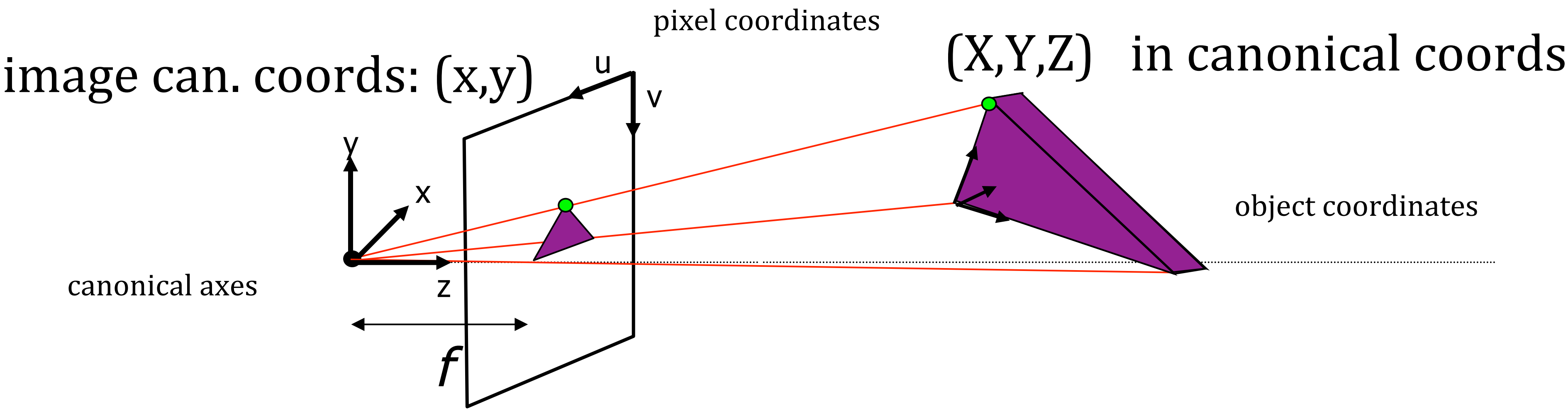


Add coordinate systems in order to describe feature points...

Coordinate Systems



From 3d to 2d



$$x = \frac{fX}{Z} \qquad y = \frac{fY}{Z}$$

a <u>nonlinear</u> transformation

goal: to recover information about (X,Y,Z) from (x,y)

Camera Calibration

- Camera Model
  - $[u\ v\ 1]$ Pixel coords
  - $\begin{bmatrix} x_w & y_w & z_w & 1 \end{bmatrix}^T$ World coords

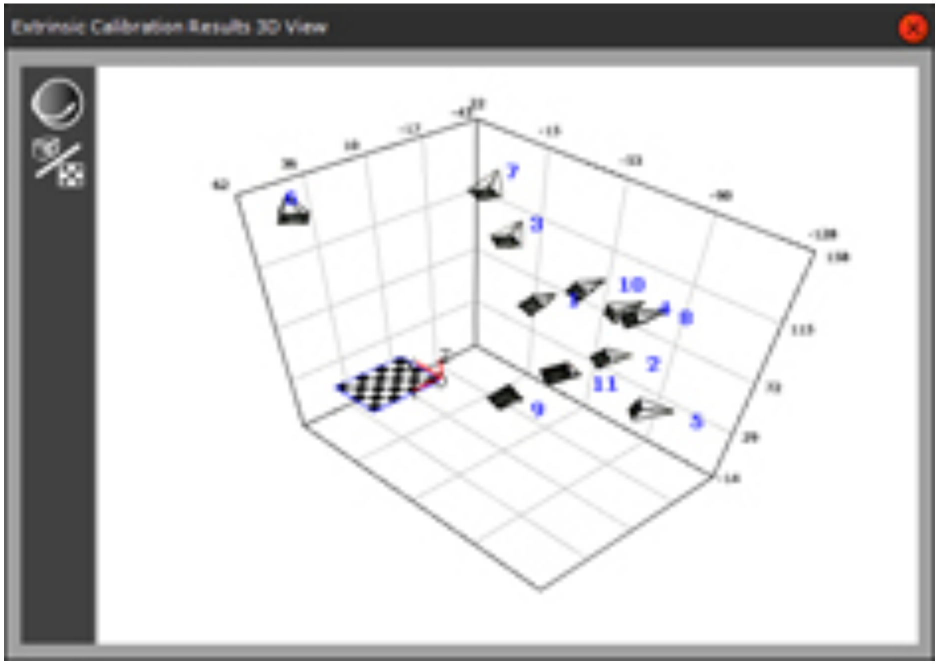
$$z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A \begin{bmatrix} R & T \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$

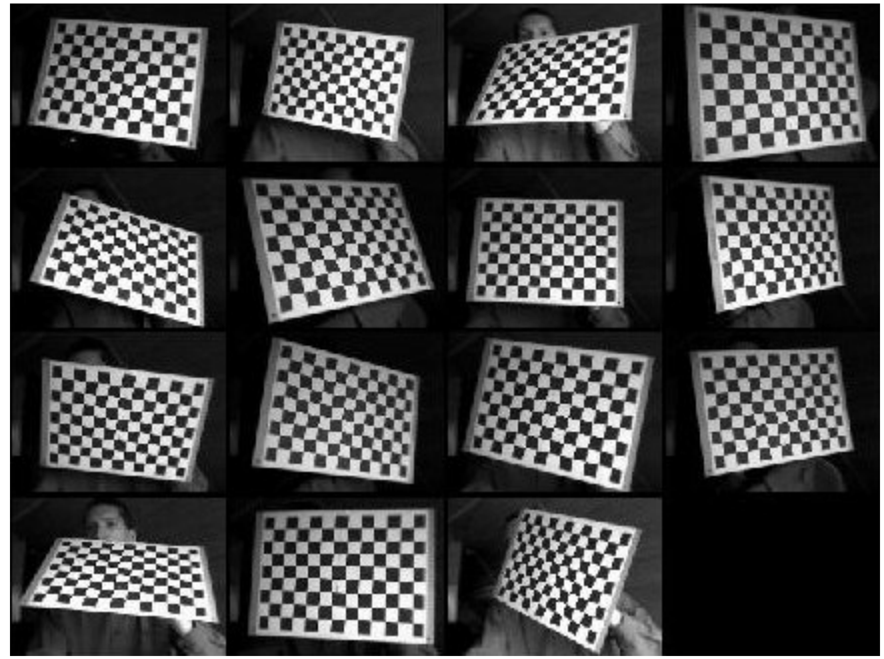
- Intrinsic Parameters
  - $\alpha_x = f \cdot m_x, \alpha_y = f \cdot m_y$ focal lengths in pixels
  - γ skew coefficient
  - u_0, v_o focal point

$$A = \begin{bmatrix} \alpha_x & \gamma & u_0 \\ 0 & \alpha_y & v_o \\ 0 & 0 & 1 \end{bmatrix}$$

- Extrinsic Parameters
  - $\begin{bmatrix} R & T \end{bmatrix}$ Rotation and Translation

Camera Calibration



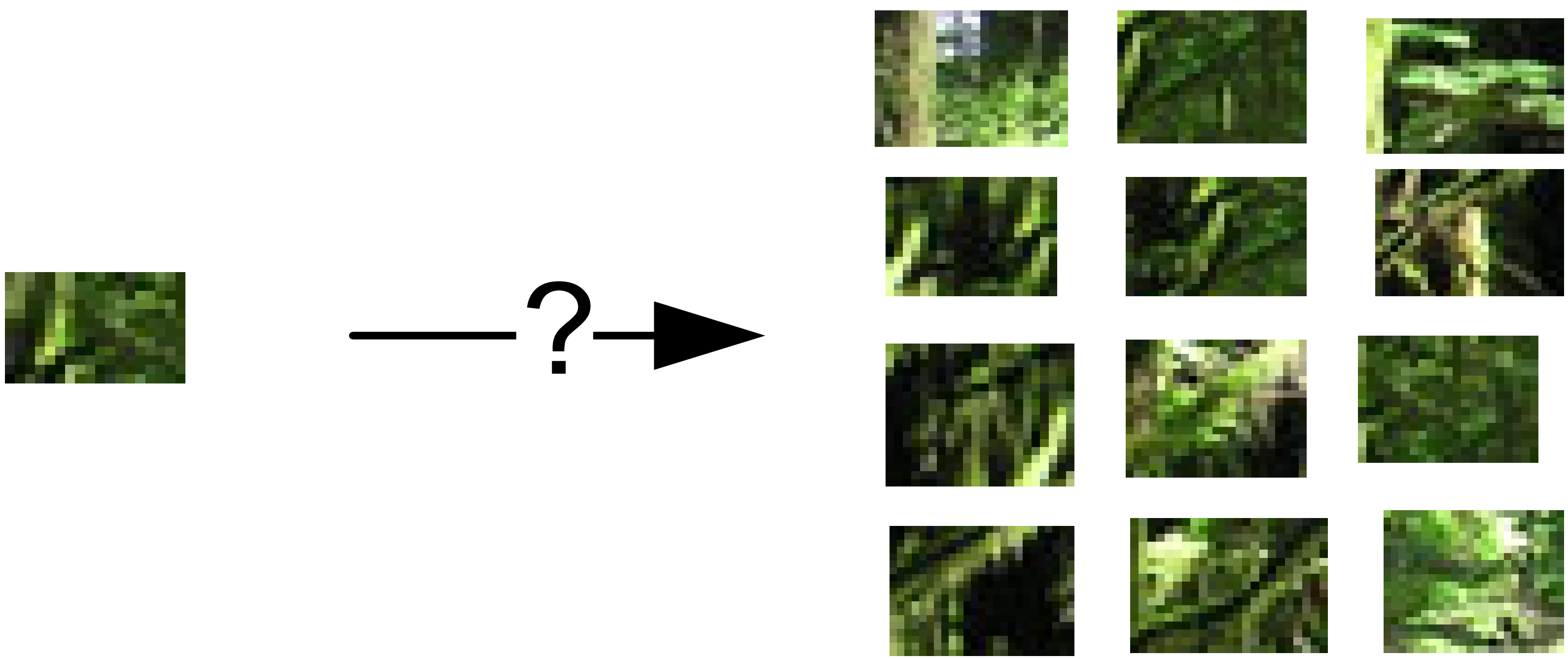


Existing packages in MATLAB, OpenCV, etc

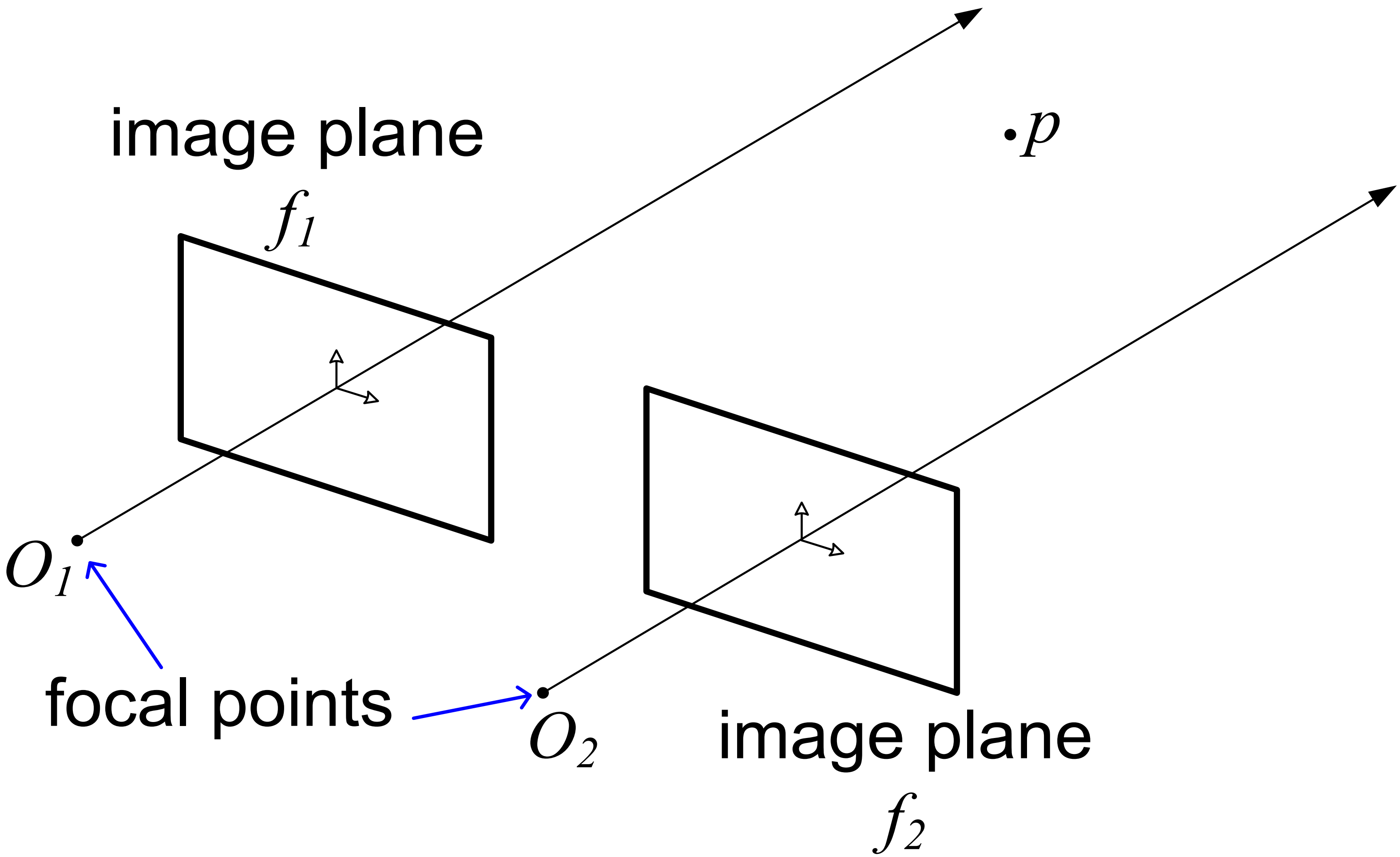
Correspondence Problem

Correspondence

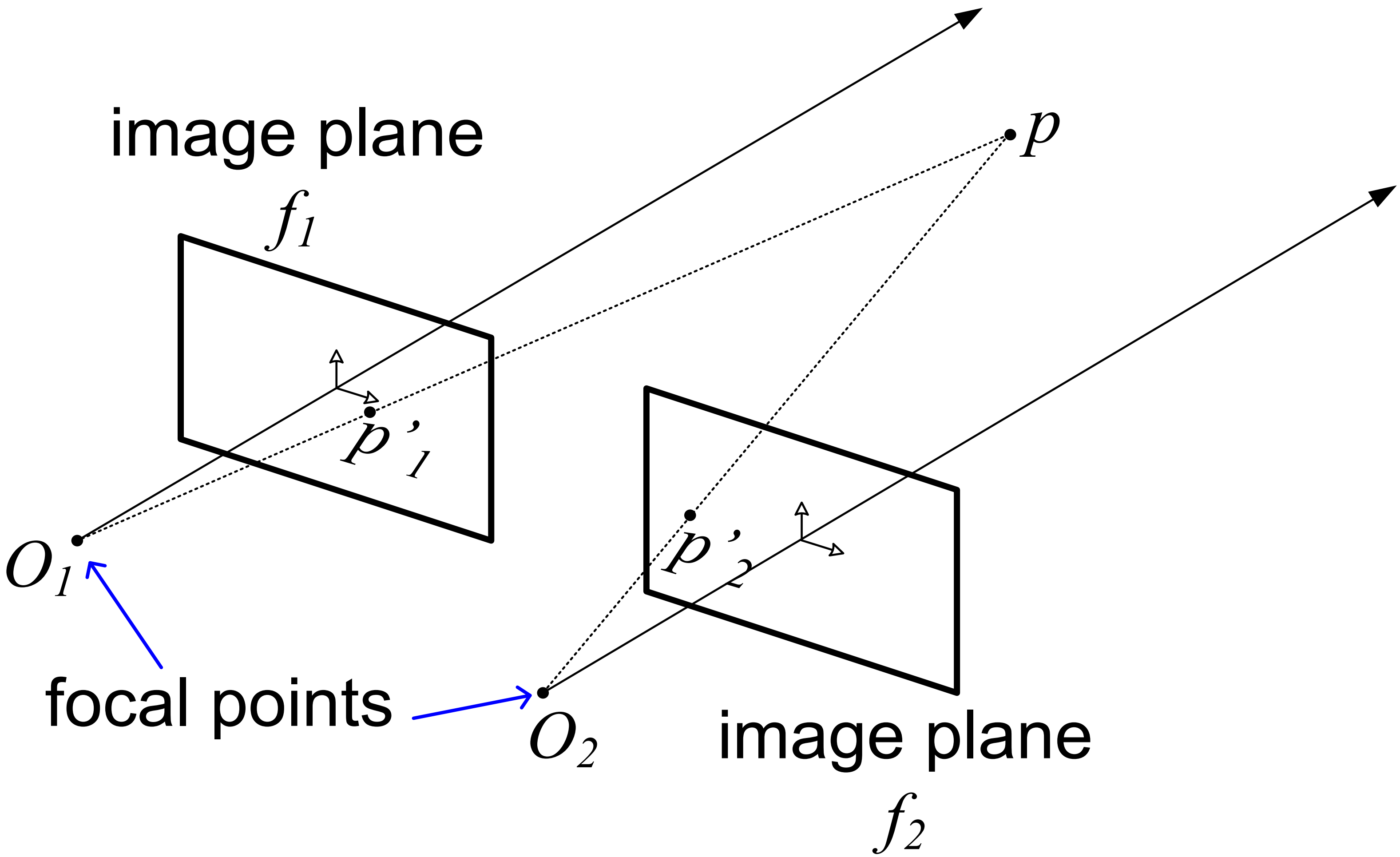
From I_1 From I_2



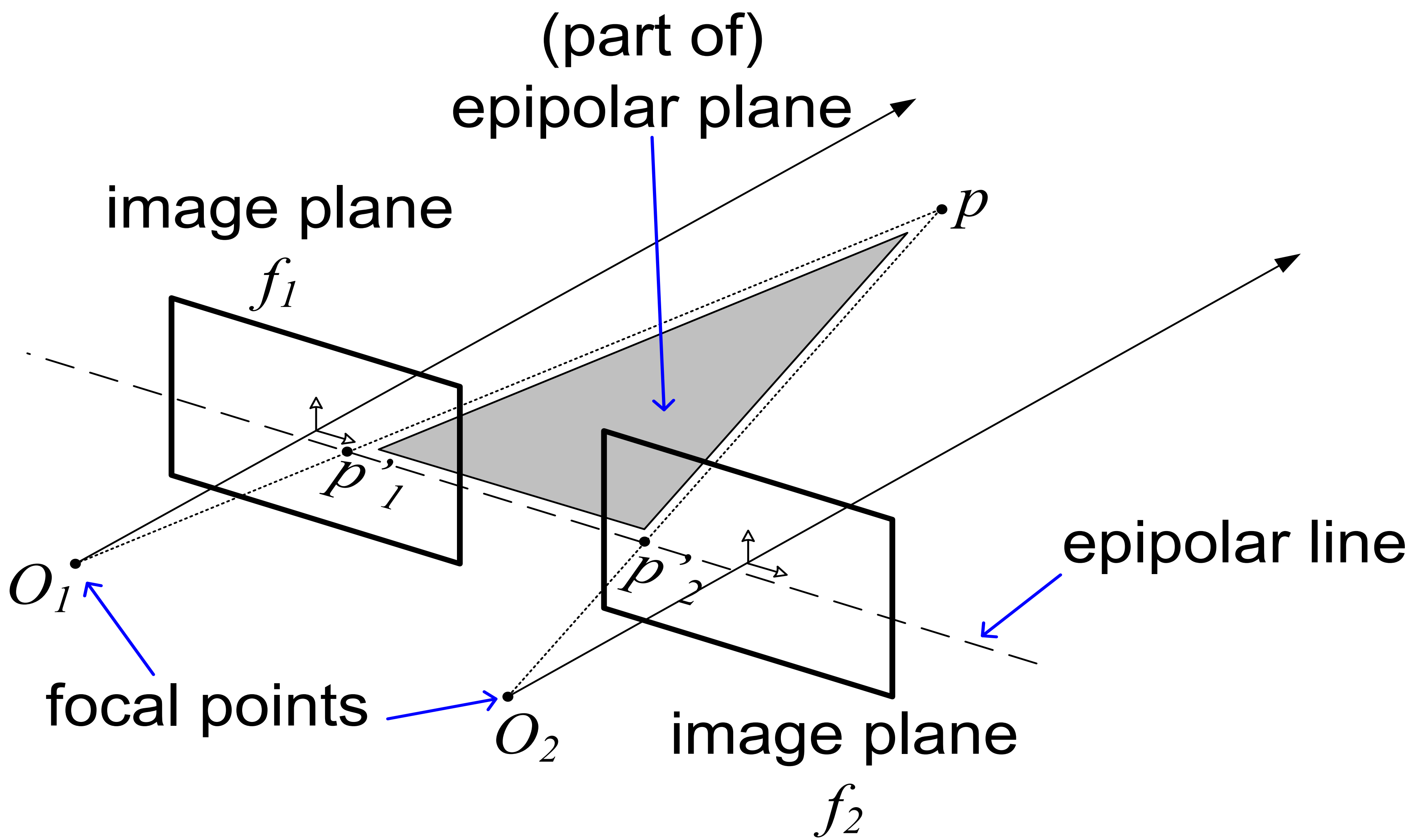
Stereo Vision: Pinhole Camera



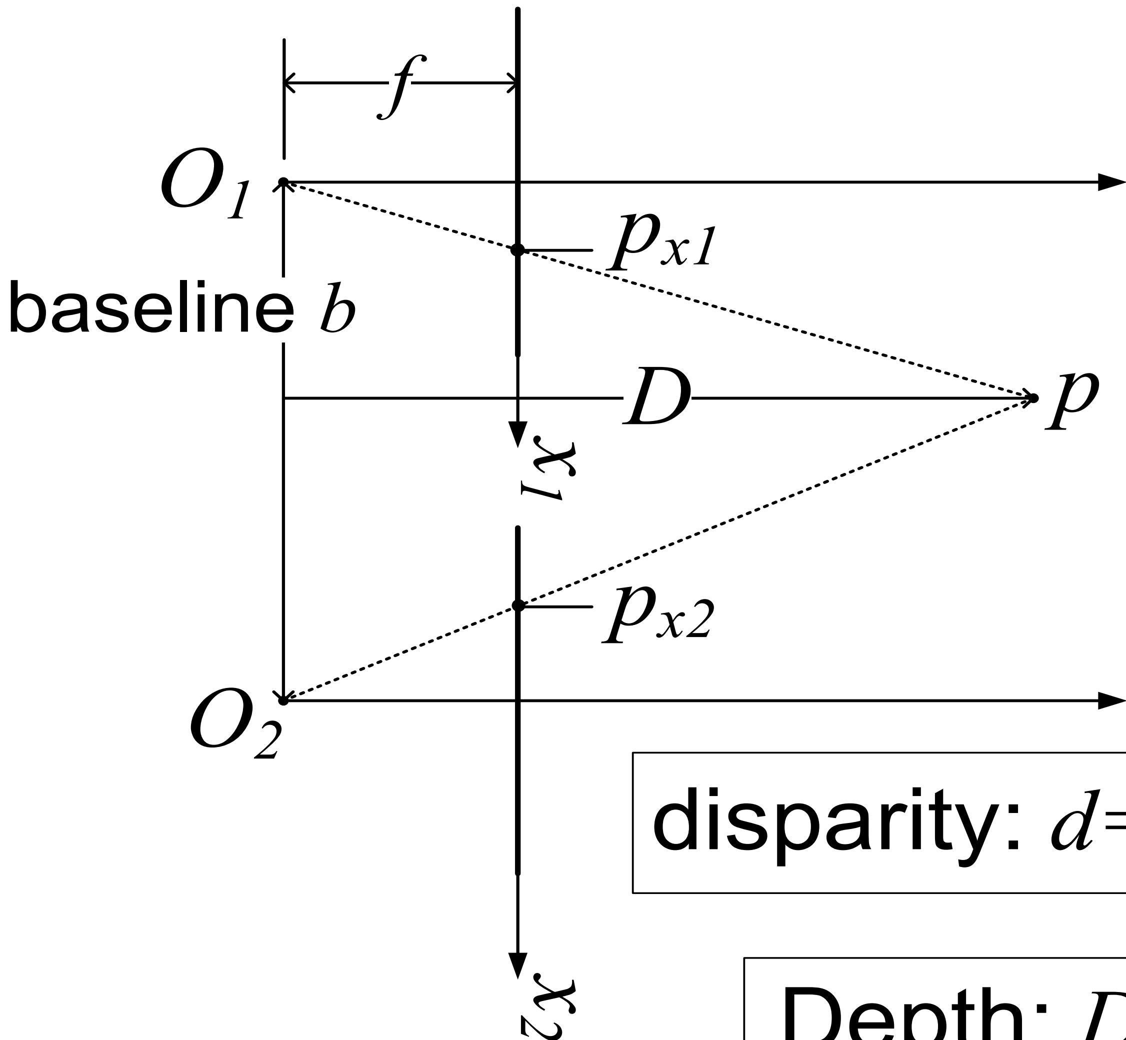
Stereo Vision: Pinhole Camera



Stereo Vision: Pinhole Camera



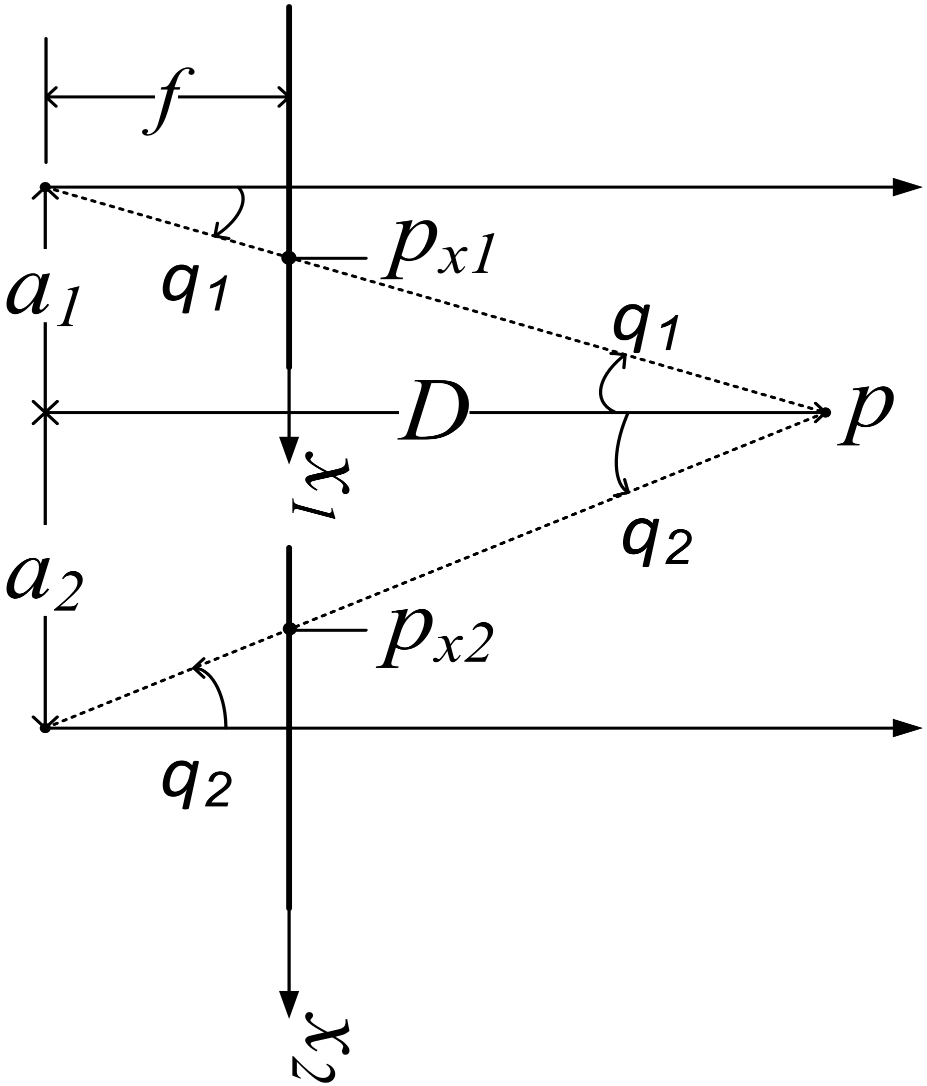
Stereo Vision: Pinhole



disparity: $d = p_{x1} - p_{x2}$

Depth: $D = fb/d$

Stereo Vision: Pinhole



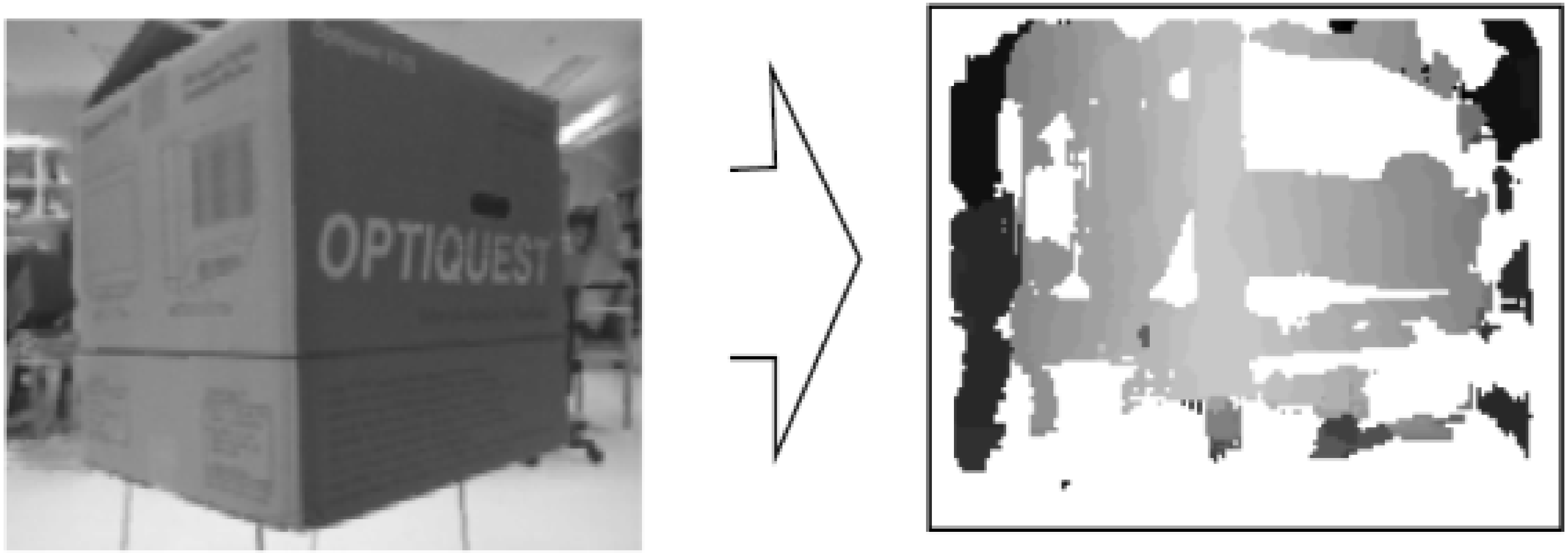
$$\frac{p_{x1}}{f} = \frac{a_1}{D}$$

$$\frac{p_{x2}}{f} = \frac{a_2}{D}$$

$$a_1 + a_2 = b$$

Large Baseline

Stereo: Disparity Map

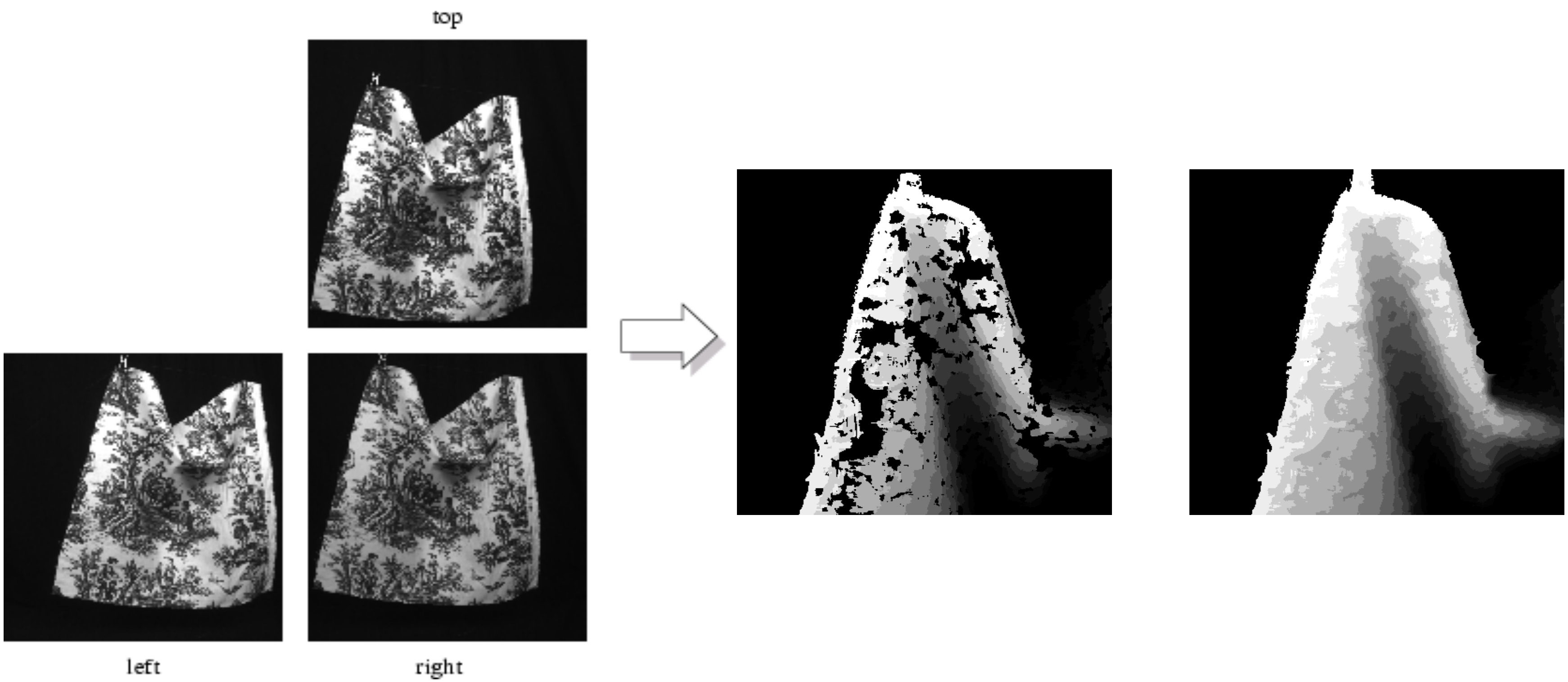


Using real-time stereo vision for mobile robot navigation

Don Murray   Jim Little

Computer Science Dept.
University of British Columbia
Vancouver, BC, Canada V6T 1Z4

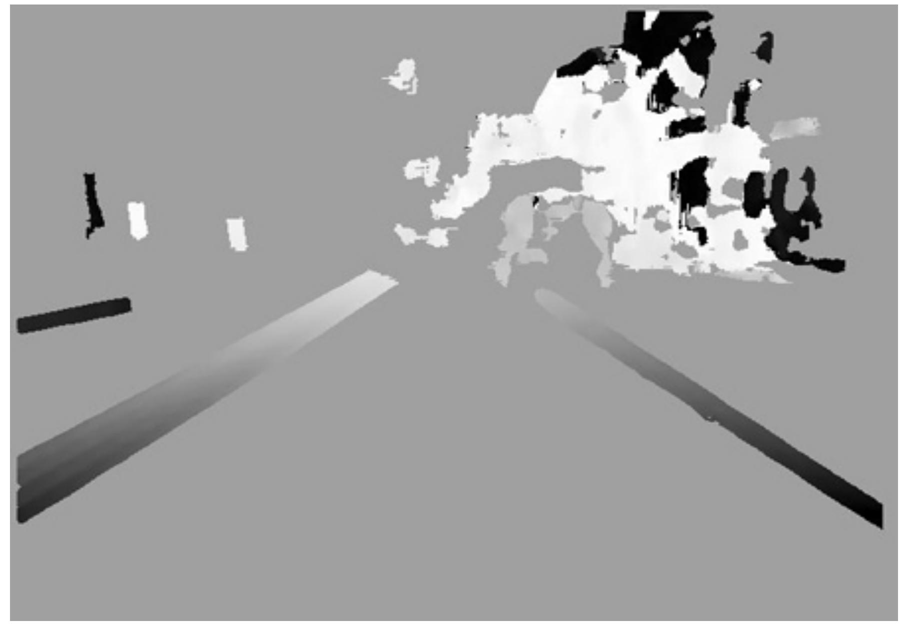
Another Example (Hole Filling)



Cloth Parameters and Motion Capture by David Pritchard

B.A.Sc., University of Waterloo, 2001

Depth Map in a City



Good Feature

- High Recall
- Good Precision
- Feature Detection
- Feature Matching
- Several Alternatives:
  - Harris Corners (OpenCV)
  - SURF (OpenCV)
  - SIFT
  - etc