NYU

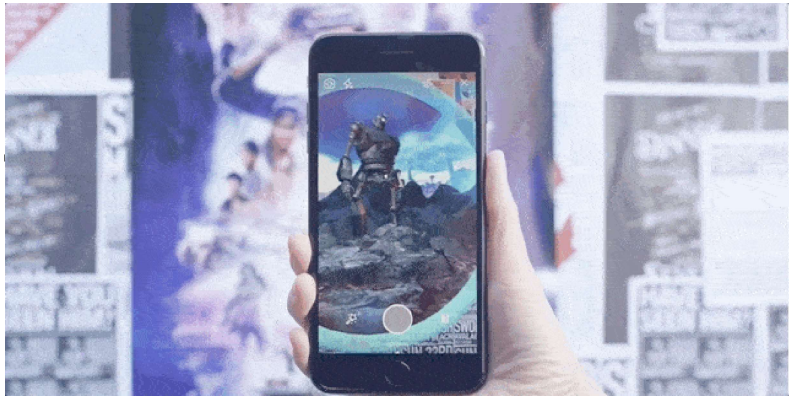
Virtual and Augmented Reality

CS-GY 9223/CUSP-GX 6004

https://nyu-icl.github.io/courses/2022fall-vr-ar

Prof. Qi Sun
qisun@nyu.edu | www.qisun.me

What are we tracking?



- Goal: track pose of headset, controller, …
- What is a pose?
    - 3D position of the tracked object
    - 3D orientation of the tracked object, e.g. using quaternions or Euler angles
- Why? So we can map the movement of our head to the motion of the camera in a virtual environment – motion parallax!

Overview of Positional Tracking

“outside-in tracking”: external sensors, cameras, or markers are required (i.e. tracking constrained to specific area)

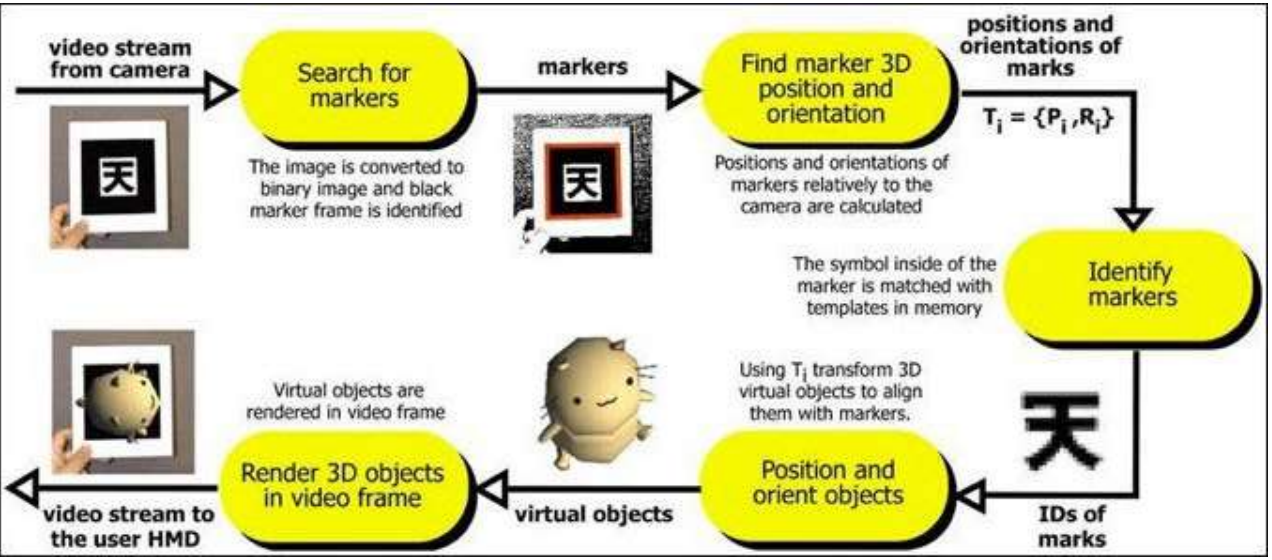
- used by most VR headsets right now, but everyone is feverishly working on insight-out tracking!

“inside-out tracking”: camera or sensor is located on HMD, no need for other external devices to do tracking

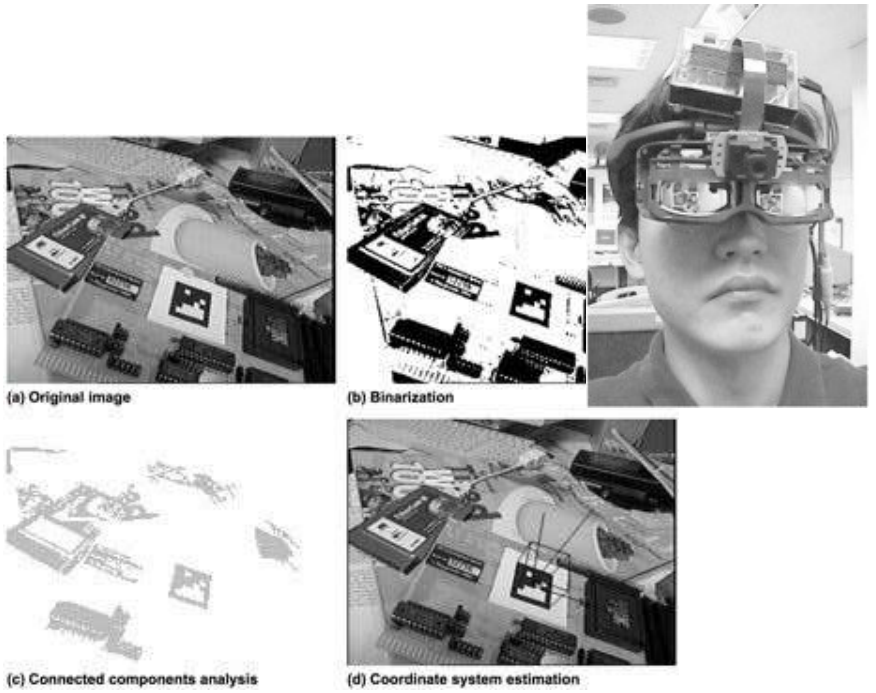
- simultaneous localization and mapping (SLAM) – classic computer & robotic vision problem (beyond this class)

Marker-based Tracking

- seminal papers by Rekimoto 1998 and Kato & Billinghurst 1999
- widely adopted after introduced by ARToolKit

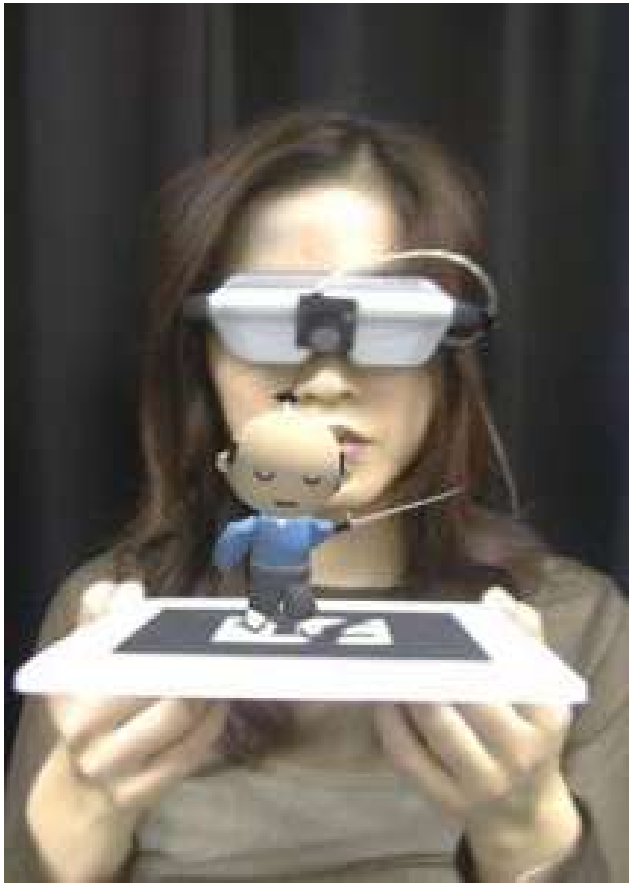


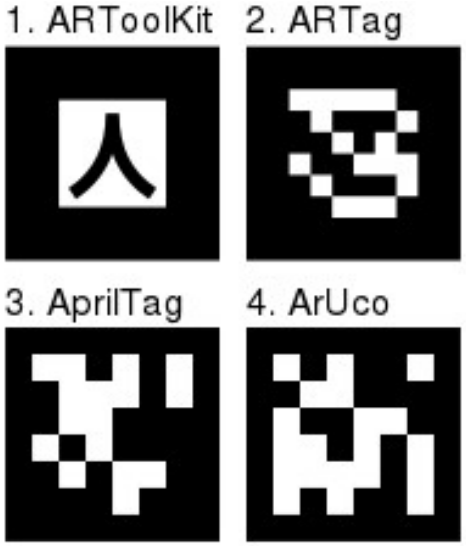
Kato, Billinghurst - ARToolKit



Rekimoto - Matrix

Marker-based Tracking





Tutorial:
https://medium.com/dscvitpune/quick-introduction-to-image-tracking-using-ar-foundation-dea7d2fbdfe9

Inside-out Tracking

Google's Project Tango

Google’s Project Tango

4MP Camera

2x Computer Vision Processors

Integrated Depth Sensing

Motion Tracking Camera

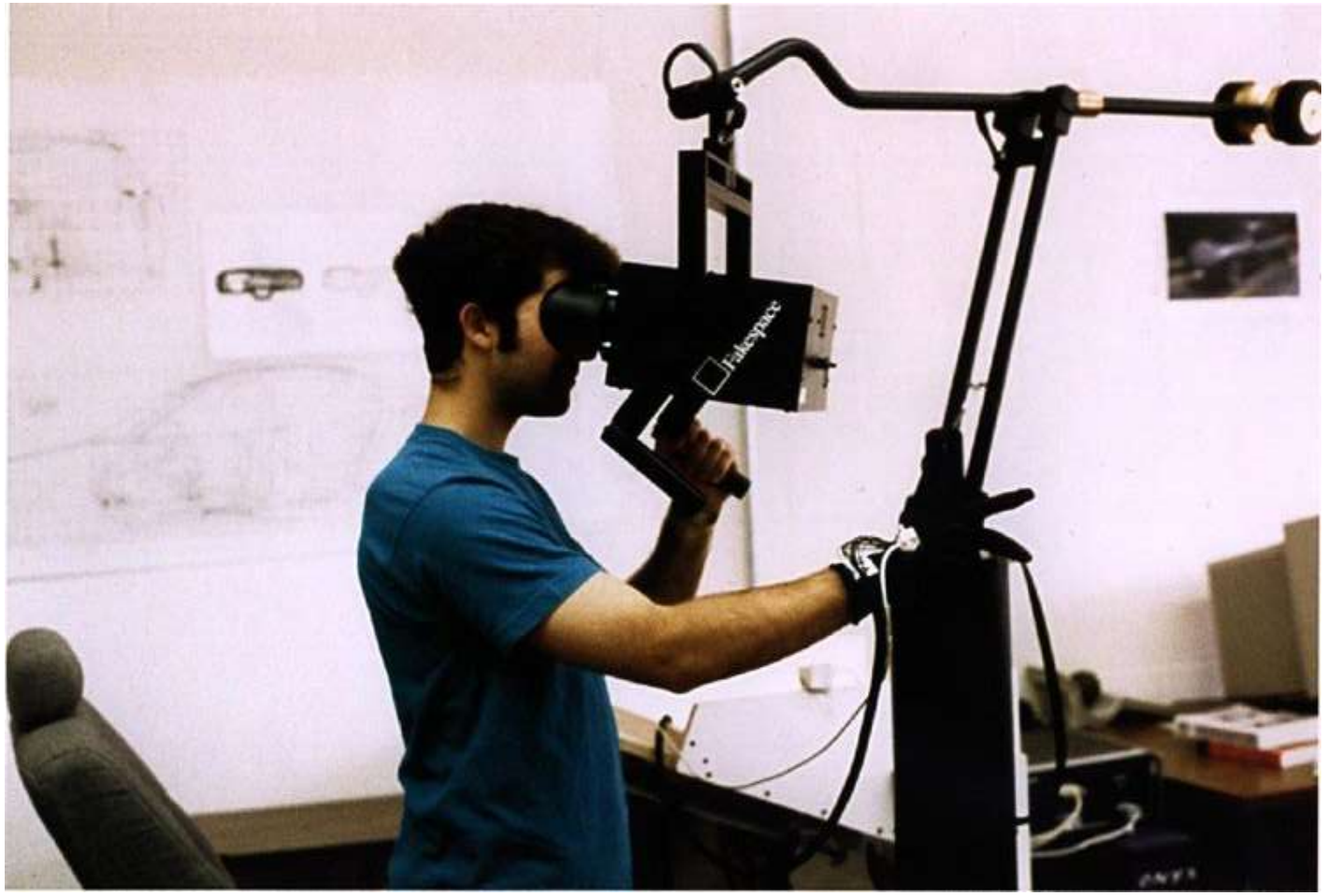
also used but not shown: IMU

problem: SLAM via sensor fusion

“Outside-in Tracking”

- mechanical tracking
- ultra-sonic tracking
- magnetic tracking
- optical tracking
- GPS
- WIFI positioning
- marker tracking
- ...

Positional Tracking - Mechanical



some mechanical linkage, e.g.

- fakespace BOOM
- microscribe

Positional Tracking - Mechanical

pros:

- super low latency
- very accurate

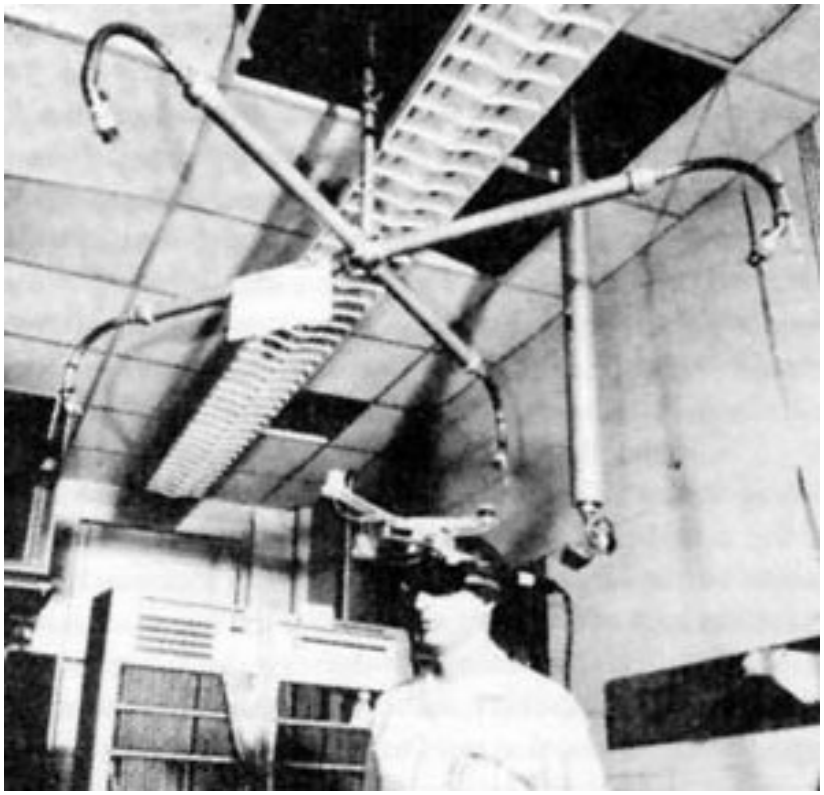
cons:

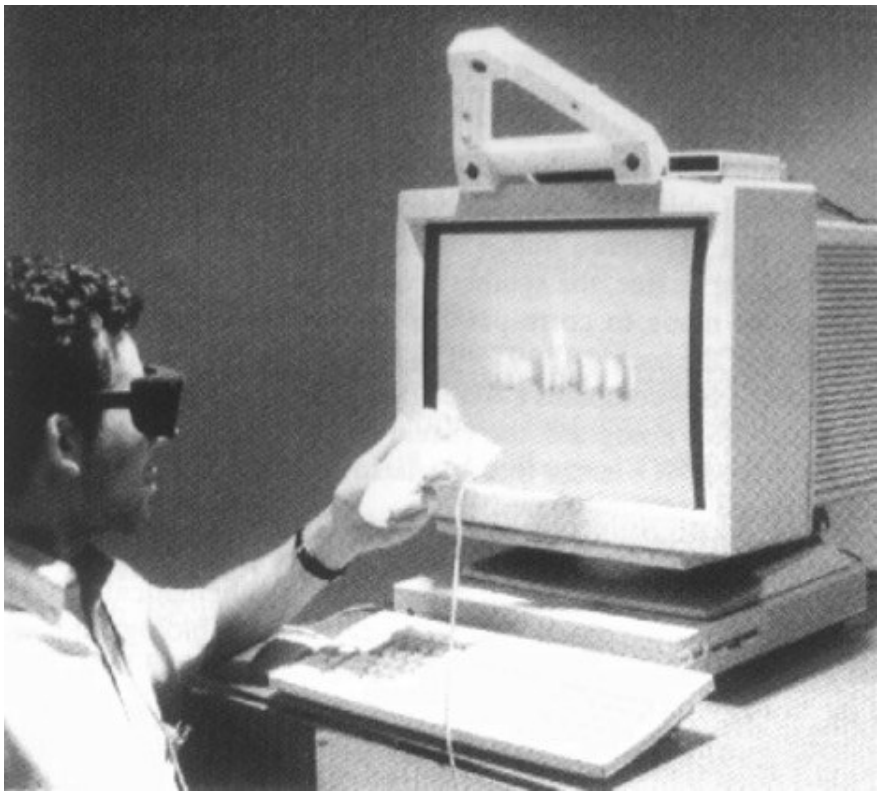
- cumbersome
- “wired” by design

Positional Tracking – Ultra-sonic

- 1 transmitter, 3 receivers ➔ triangulation

Ivan Sutherland's "Ultimate Display"





Logitech 6DOF

Positional Tracking – Ultra-sonic

pros:

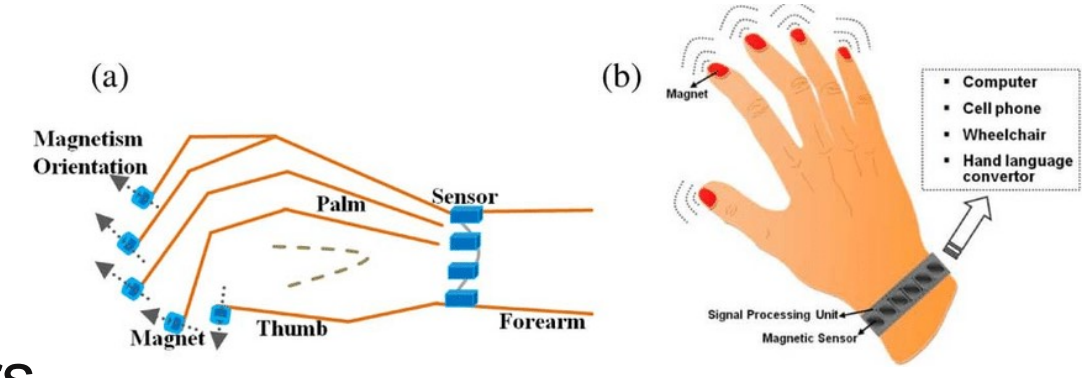
- can be light, small, inexpensive

cons:

- line-of-sight constraints
- susceptible to acoustic interference
- low update rates

Positional Tracking - Magnetic

- reasonably good accuracy
- position and orientation
- 3 axis magnetometer in sensors
- need magnetic field generator (AC, DC, …), e.g. Helmholtz coil
- magnetic field has to oscillate and be sync'ed with magnetometers



Positional Tracking - Magnetic

pros:

- small, low cost, low latency sensors
- no line-of-sight constraints

3 axis Helmholtz coil
www.directvacuum.com

cons:

- somewhat small working volume
- susceptible to distortions of magnetic field
- not sure how easy it is to do this untethered (need to sync)

Positional Tracking - Magnetic

Magic Leap One controller tracking:

- magnetic field generator in controller
- magnetometer in headset

https://www.ifixit.com/Teardown/Magic+Leap+One+Teardown/112245

Positional Tracking - Optical

- track active (near IR) LEDs with cameras

OR

- track passive retro-reflectors with IR illumination around camera
- both Oculus Rift and HTC Vive come with optical tracking

Oculus Rift

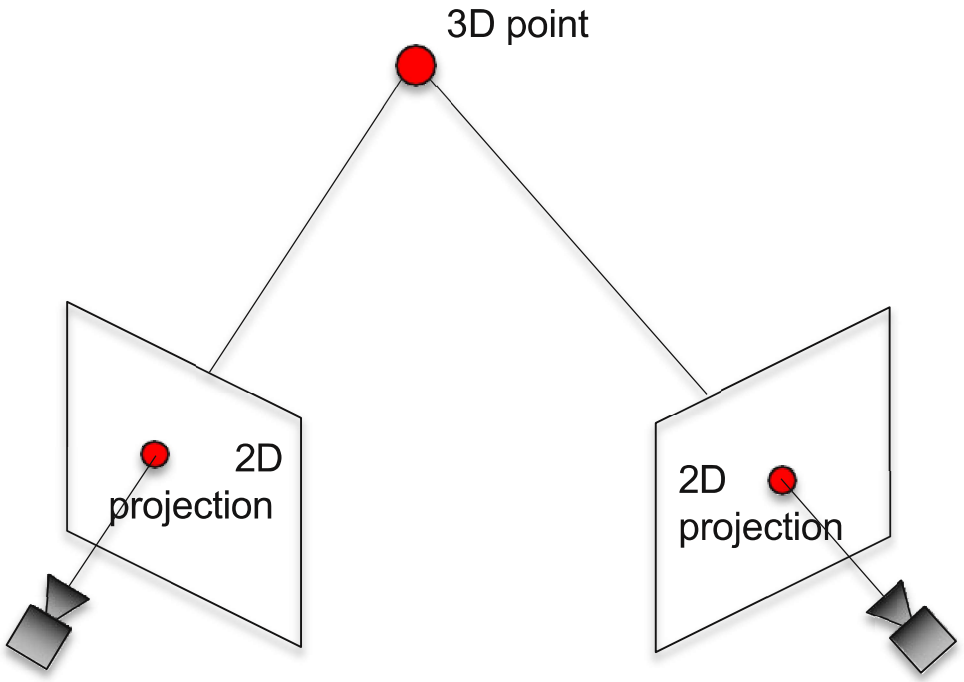
https://www.ifixit.com/Teardown/Oculus+Rift+CV1+Teardown/60612


http://steam3.com/make-magic-2015/

Understanding Pose Estimation - Triangulation

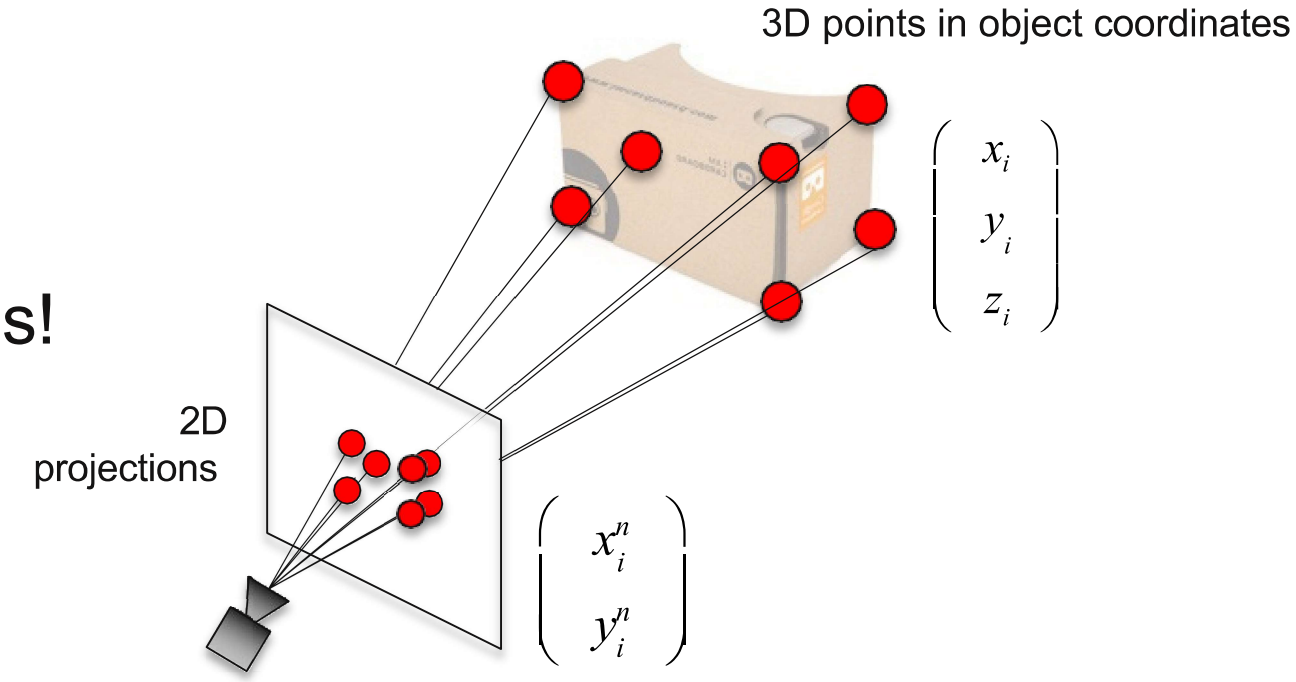
- for tracking individual 3D points, multi-camera setups usually use triangulation

- this does not give us the pose (rotation & translation) of camera or object yet



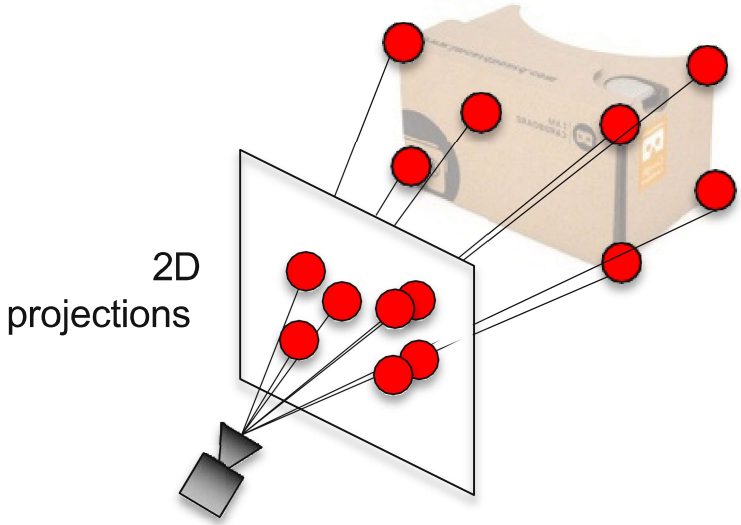
Understanding Pose Estimation

- for pose estimation, need to track multiple points with known relative 3D coordinates!



Understanding Pose Estimation

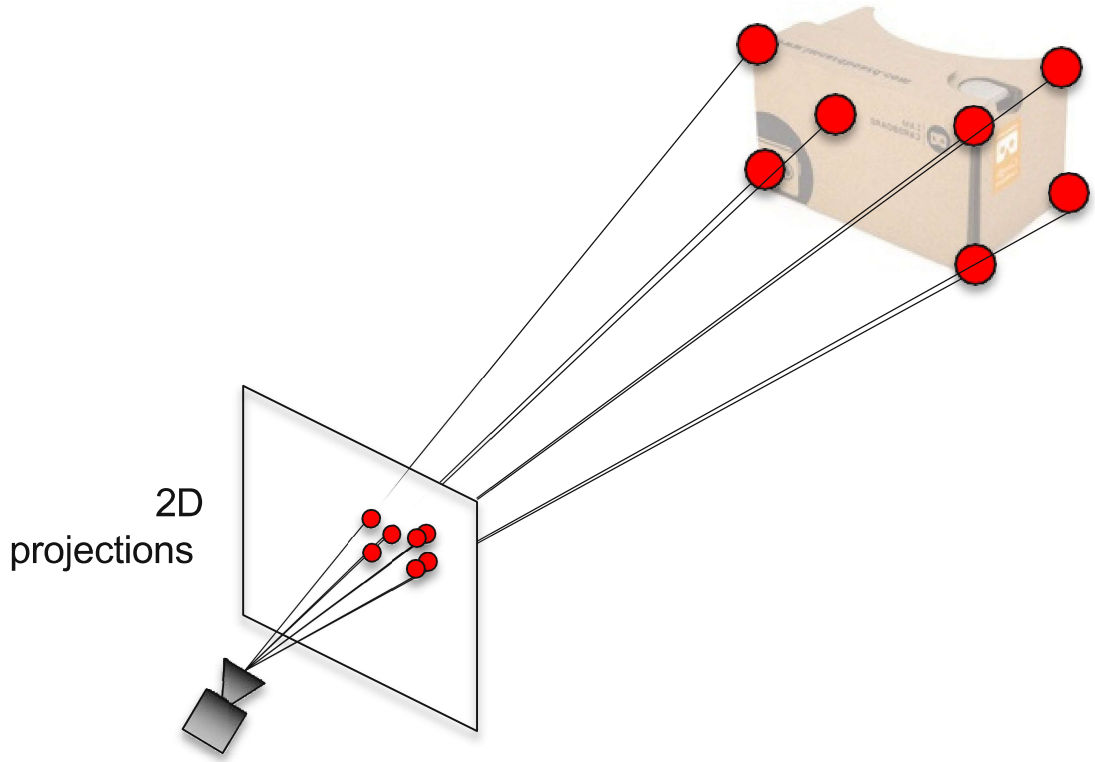
- when object is closer, projection is bigger



Understanding Pose Estimation

- when object is father, projection is smaller

... and so on ...

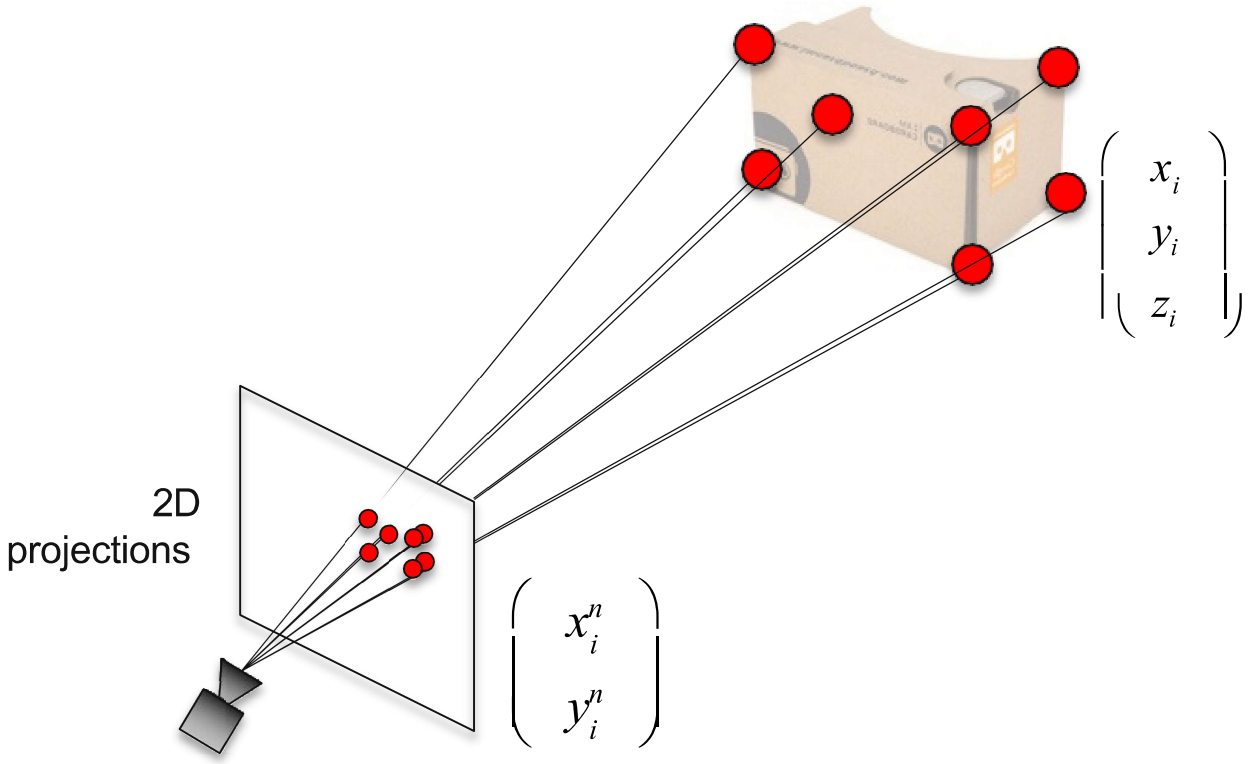


Estimating 6-DoF pose from 2D projections is known as the

Perspective-n-point problem!

Understanding Pose Estimation

1. how to get projected 2D coordinates?
2. image formation
3. estimate pose with linear homography method
4. estimate pose with nonlinear Levenberg-Marquardt method



Understanding Pose Estimation

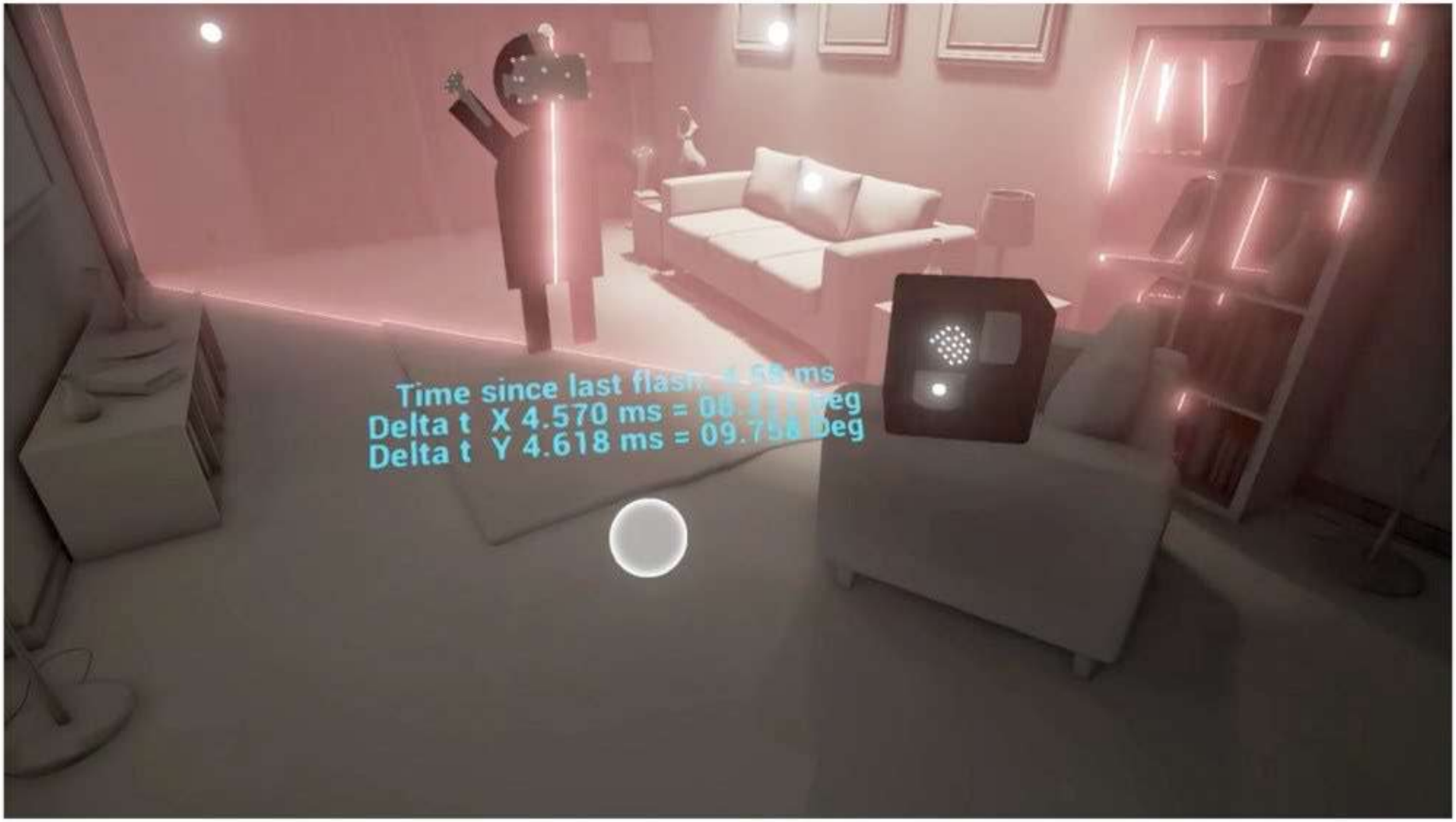
1. how to get projected
   2D coordinates?

2. Image formation

3. Estimate pose with linear homography method

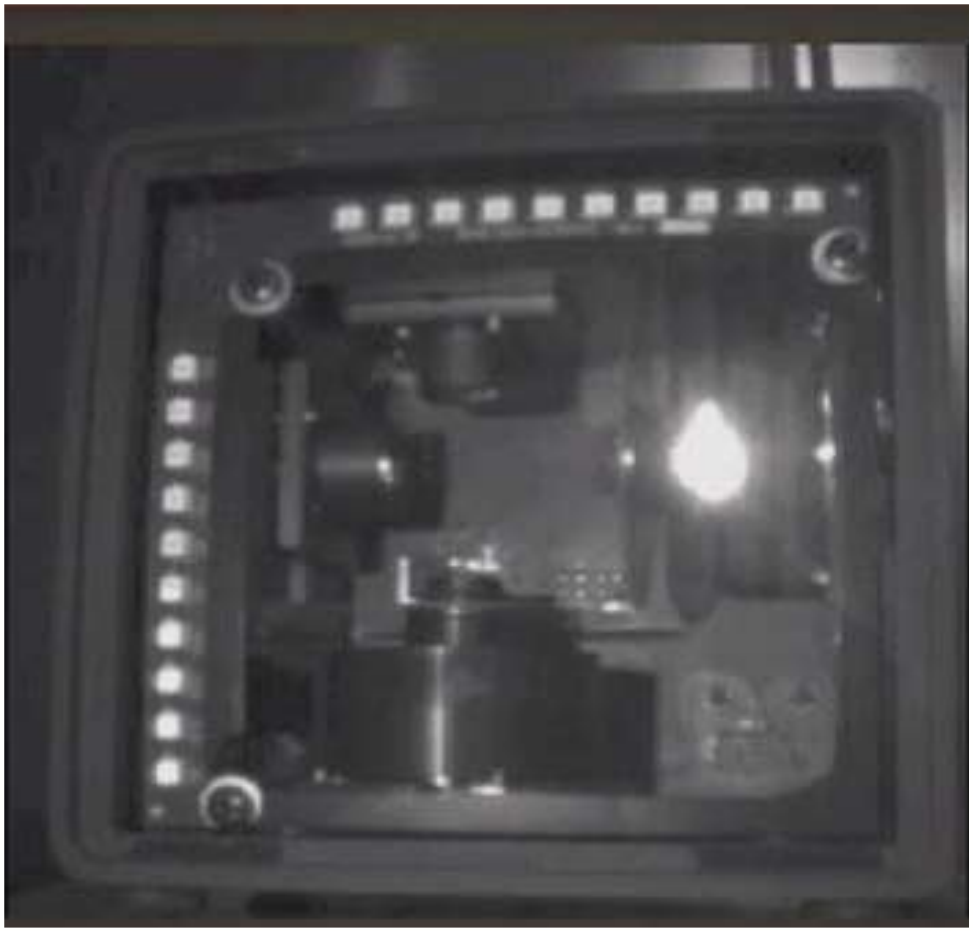
4. Estimate pose with nonlinear Levenberg- Marquardt method

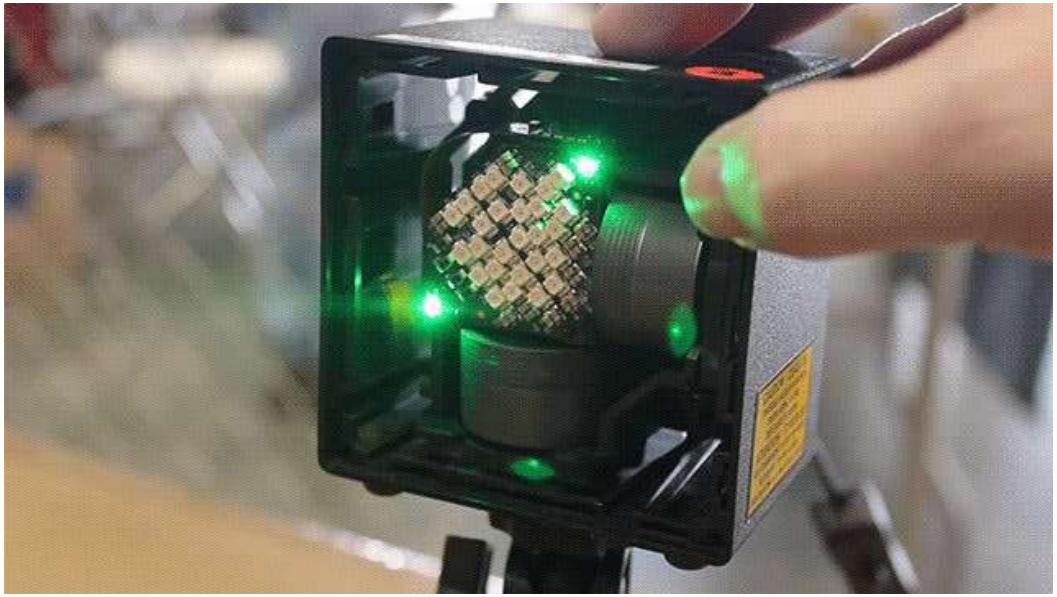
HTC Lighthouse



https://www.youtube.com/watch?v=J54dotTt7k0

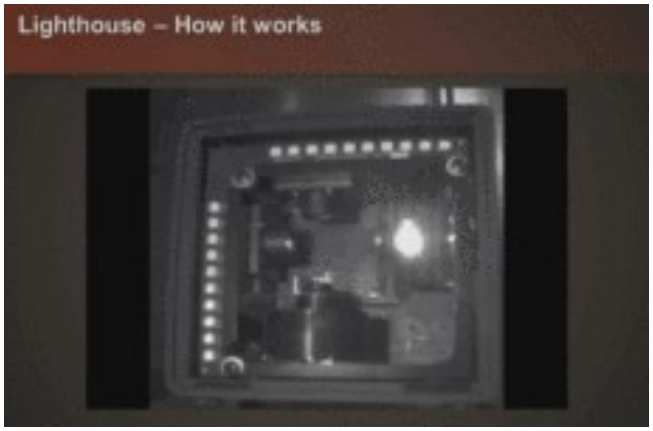
HTC Lighthouse – Base Station





http://gizmodo.com/this-is-how-valve-s-amazing-lighthouse-tracking-technol-1705356768

HTC Lighthouse – Base Station



<u>important specs</u>:

- runs at 60 Hz
    - i.e. horizontal & vertical update combined 60 Hz
    - broadband sync pulses in between each laser sweep (i.e. at 120 Hz)
- each laser rotates at 60 Hz, but offset in time
- useable field of view: 120 degrees

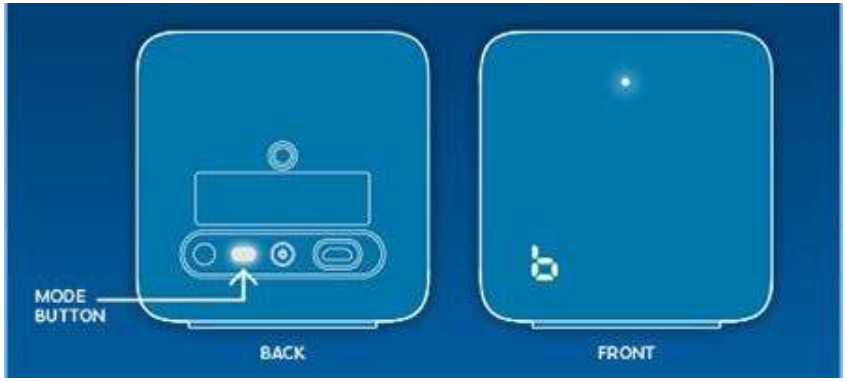
HTC Lighthouse – Base Station

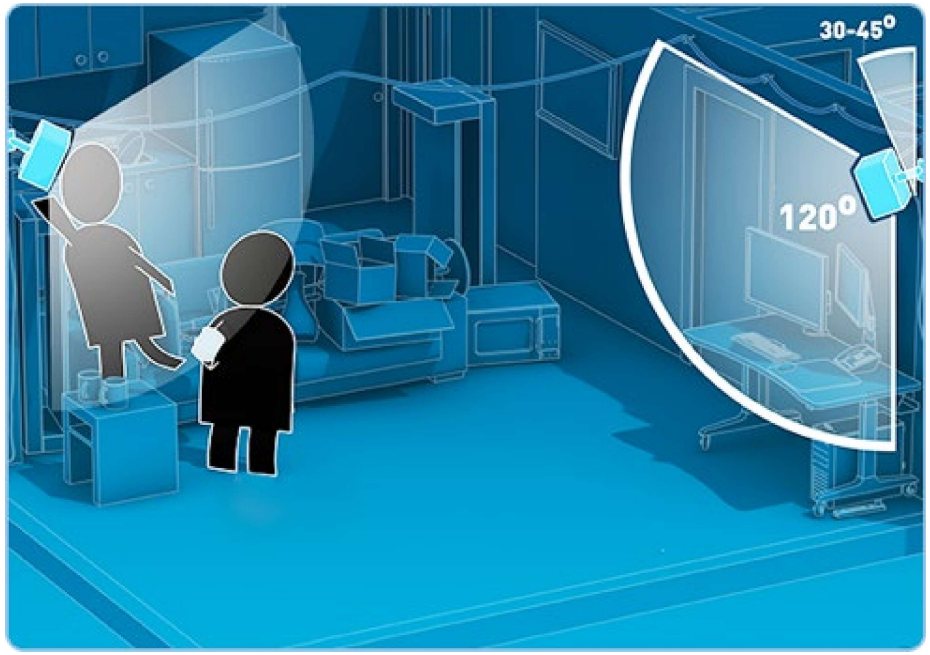
- can use up to 2 base stations simultaneously via *time-division multiplexing* (TDM)
- base station modes:

    A: TDM slave with cable sync

    B: TDM master

    C: TDM slave with optical sync

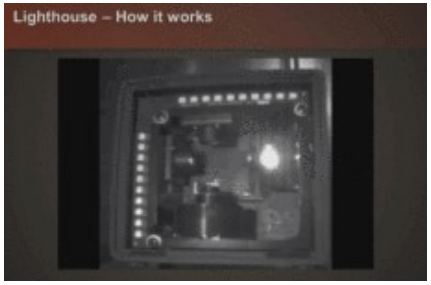




HTC

HTC Lighthouse – Base Station

- sync pulse periodically emitted (120 times per second)
- each sync pulse indicates beginning of new sweep
- length of pulse also encodes additional 3 bits of information:



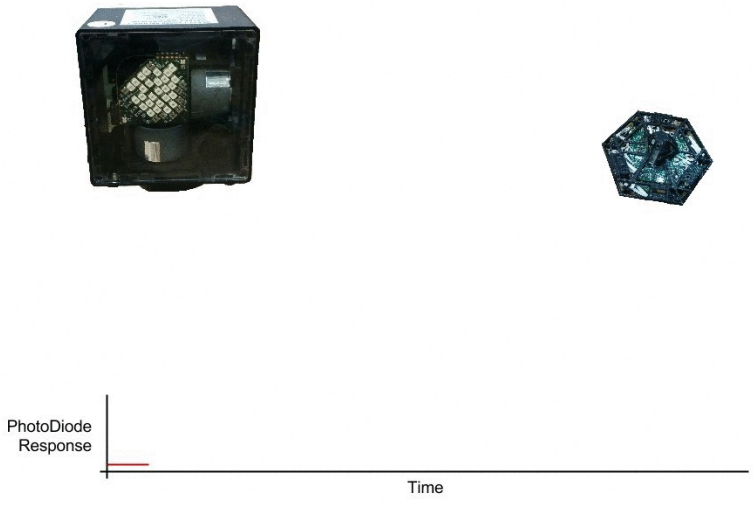
- axis: horizontal or vertical sweep to follow
- skip: if 1, then laser is off for following sweep
- data: data bits of consecutive pulses yield OOTX frame

Name	skip	data	axis	length (ticks)	length (µs)
j0	0	0	0	3000	62.5
k0	0	0	1	3500	72.9
j1	0	1	0	4000	83.3
k1	0	1	1	4500	93.8
j2	1	0	0	5000	104
k2	1	0	1	5500	115
j3	1	1	0	6000	125
k3	1	1	1	6500	135

https://github.com/nairol/LighthouseRedox/blob/master/docs/Light%20Emissions.md#sync-pulse

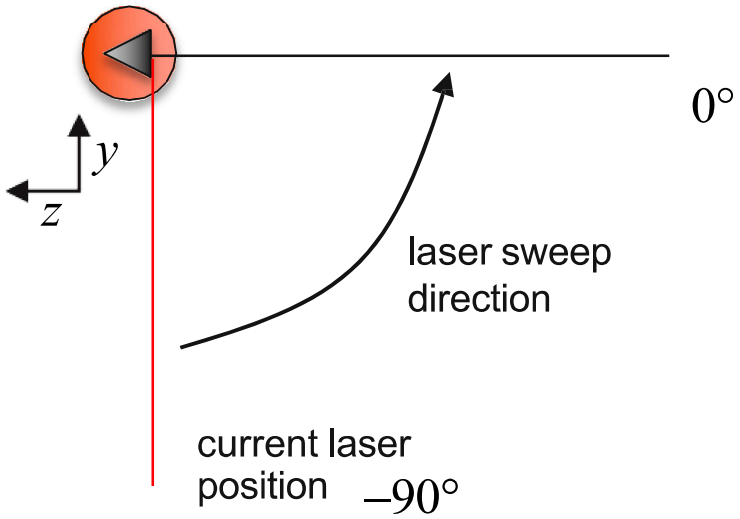
How to Get the 2D Coordinates?

- at time of respective sync pulse, laser is at 90° horizontally and -90° vertically
- each laser rotates 360° in 1/60 sec



More Details: www.Hizook.com

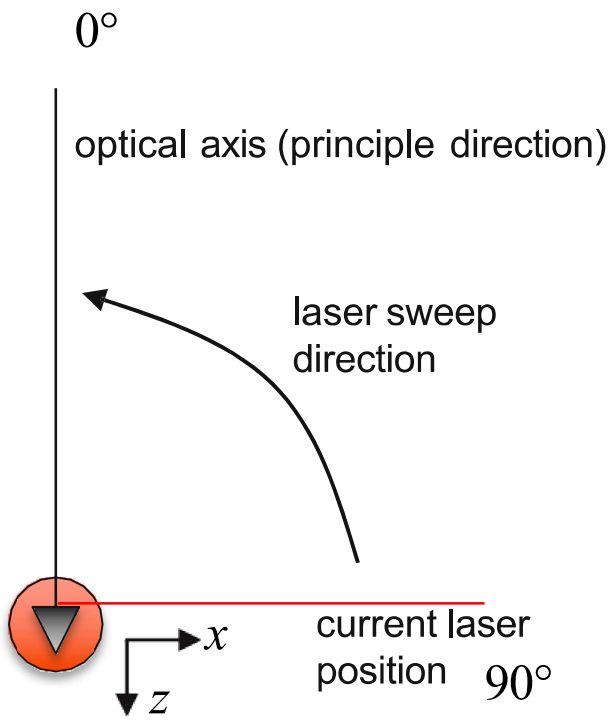
Side View



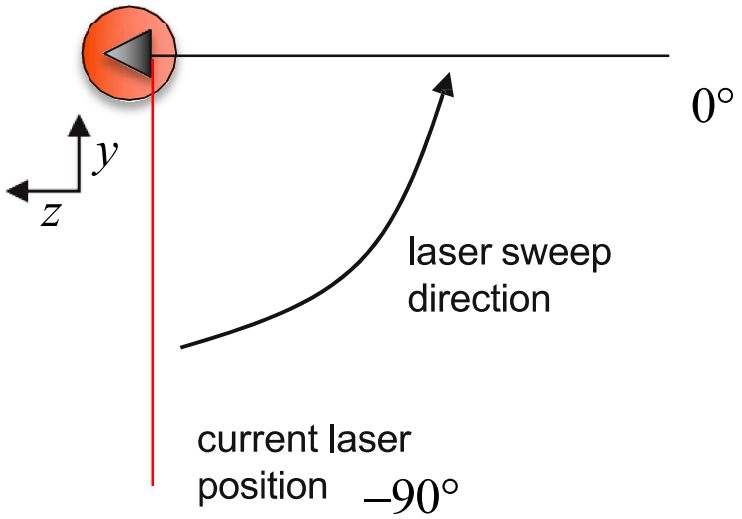
How to Get the 2D Coordinates?

- at time of respective sync pulse, laser is at 90° horizontally and -90° vertically
- each laser rotates 360° in 1/60 sec

Top View



Side View

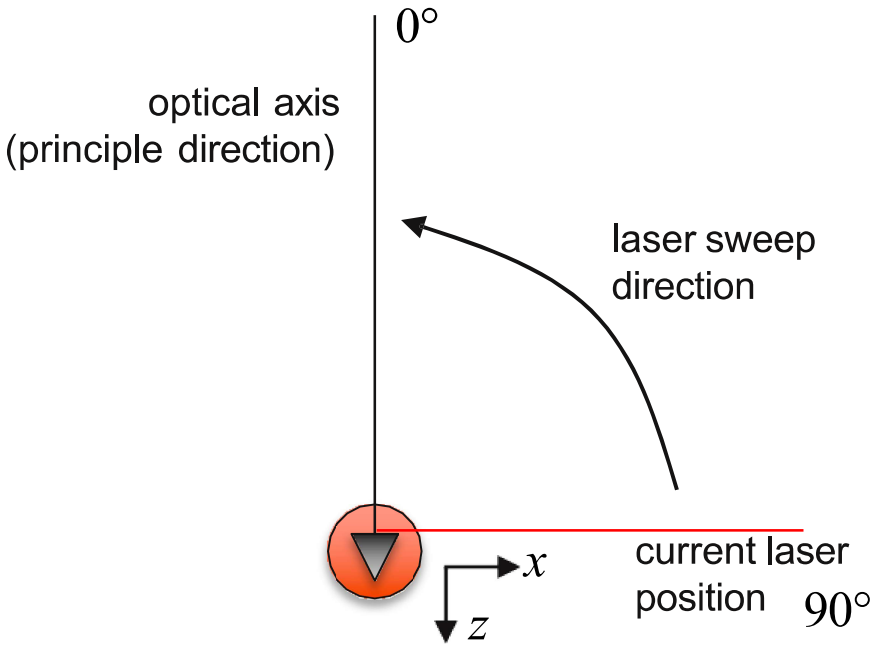


How to Get the 2D Coordinates?

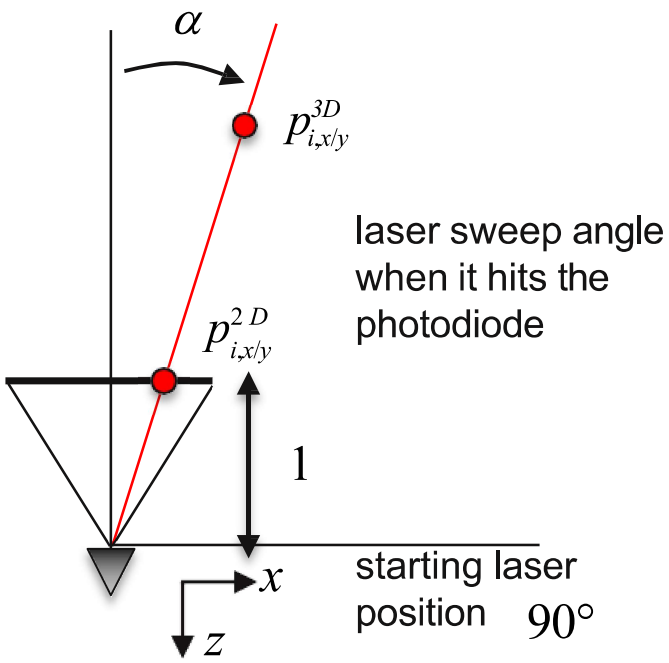
- at time of respective sync pulse, laser is at 90° horizontally and -90° vertically
- each laser rotates 360° in 1/60 sec

- convert from ticks to angle first and then to relative position on plane at unit distance

Top View



Top View



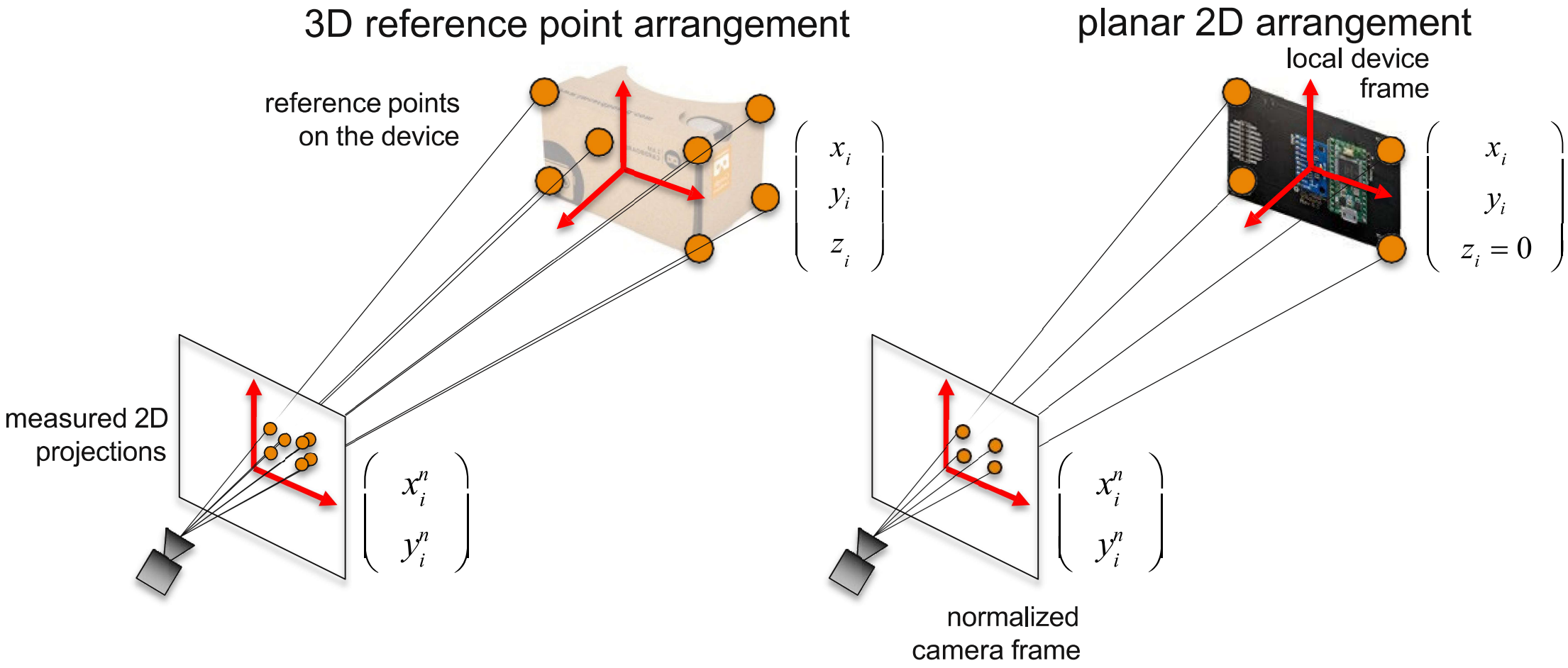
Understanding Pose Estimation

1. how to get projected 2D coordinates?
2. image formation
3. estimate pose with linear homography method
4. estimate pose with nonlinear Levenberg-Marquardt method

- how 3D points project into 2D coordinates in a camera (or a Lighthouse base station)
- very similar to graphics pipeline

Image Formation

- image formation is model for mapping 3D points in local “object” coordinate system to 2D points in “window” coordinates

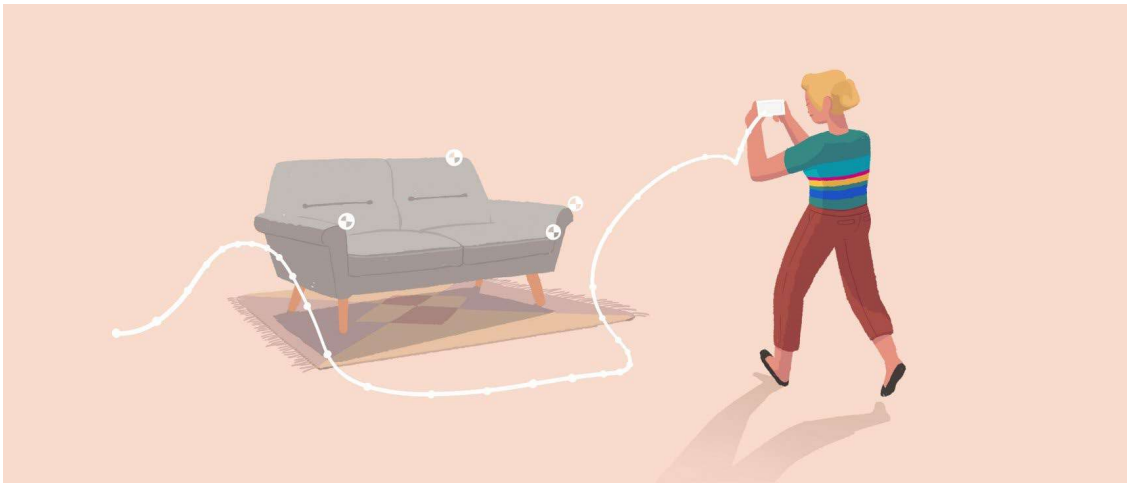


Inside-out Tracking

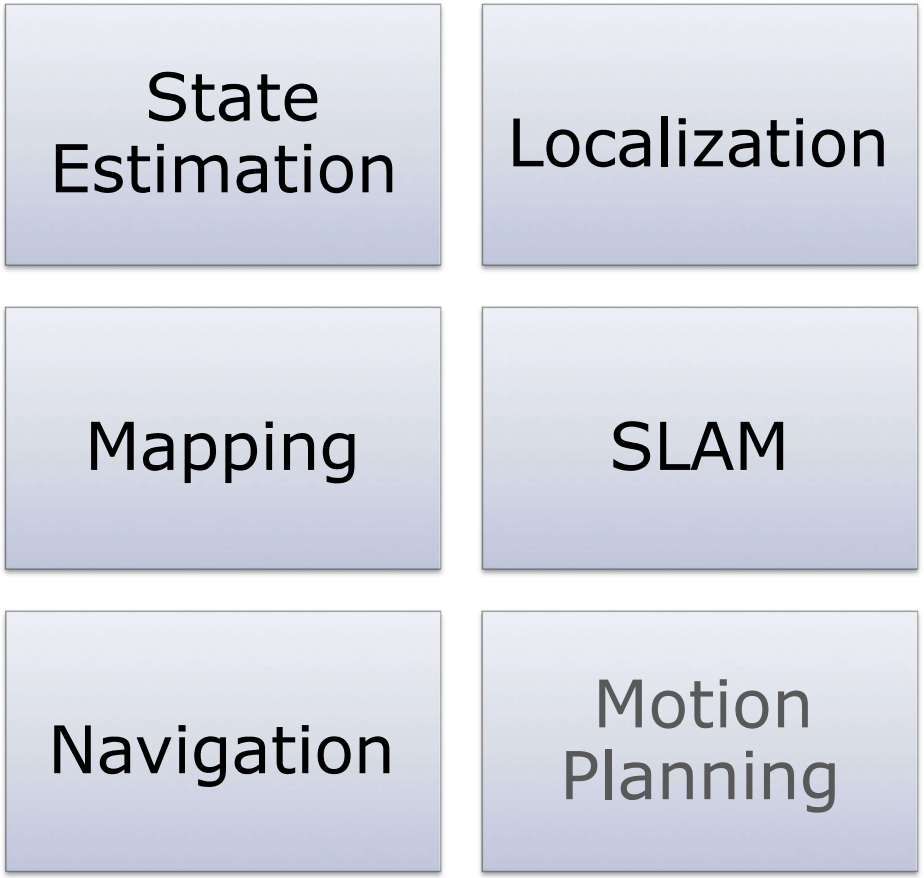
- marker-less inside-out tracking used by Microsoft HoloLens, Oculus Quest, Magic Leap, …
- eventually required by all untethered VR/AR systems

Simultaneous **L**ocalization **A**nd **M**apping

- **AR Camera** –moves through the environment
- **Mapping** – modeling the environment



Related Terms

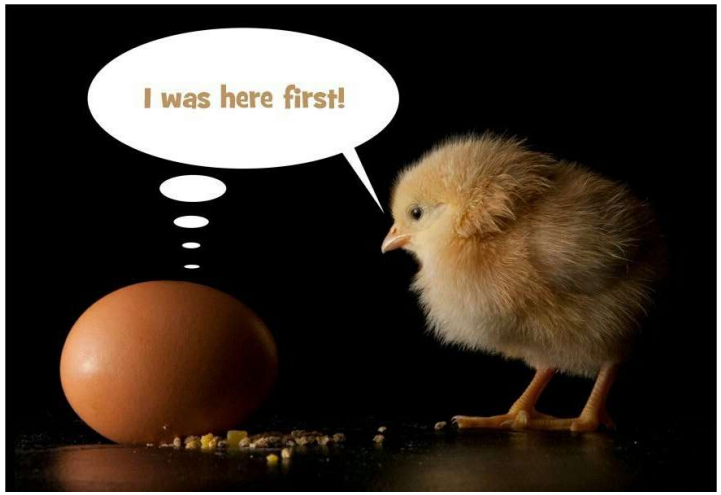


What is SLAM?

Localization: estimating the camera's location

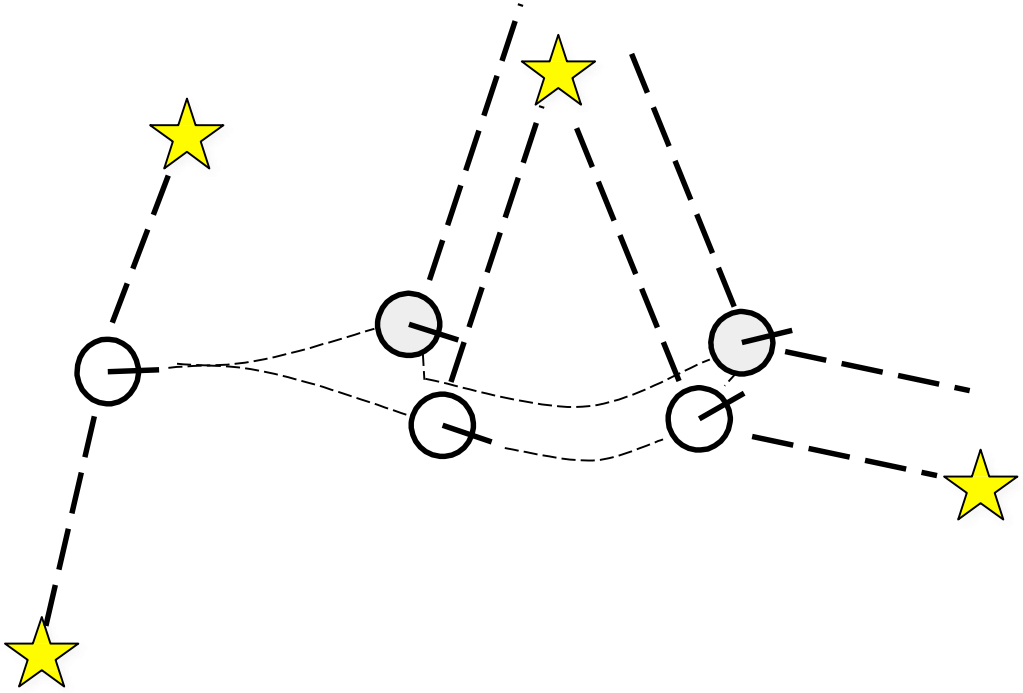
Mapping: building a map

SLAM: building a map and localizing the camera simultaneously



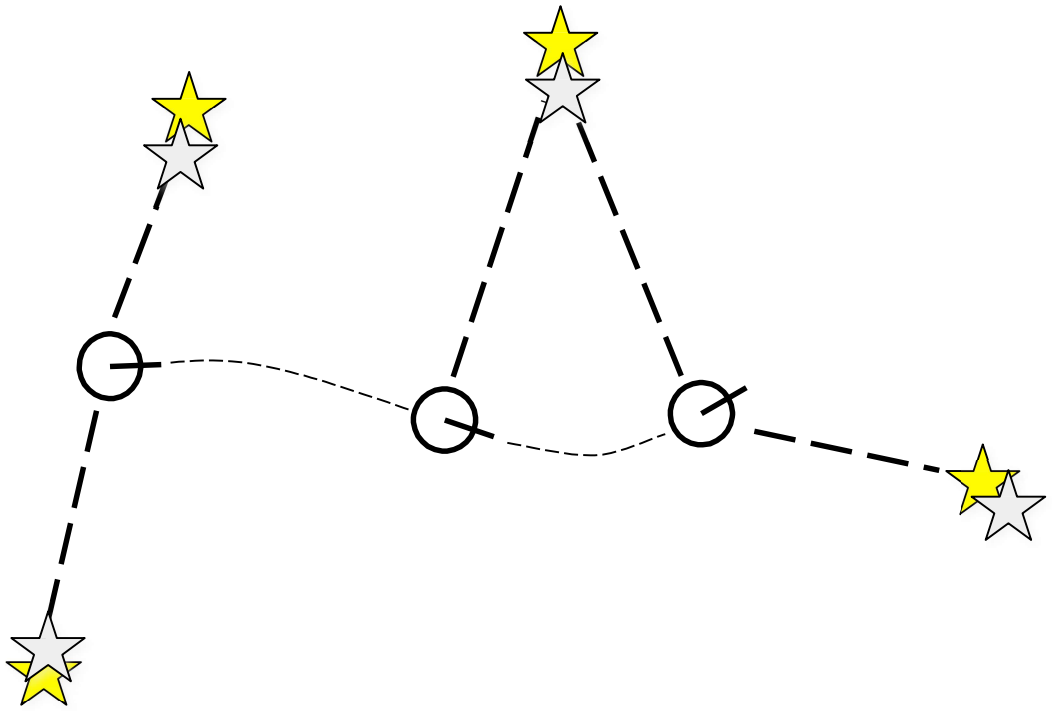
Localization Example

Estimate the camera's poses given landmarks



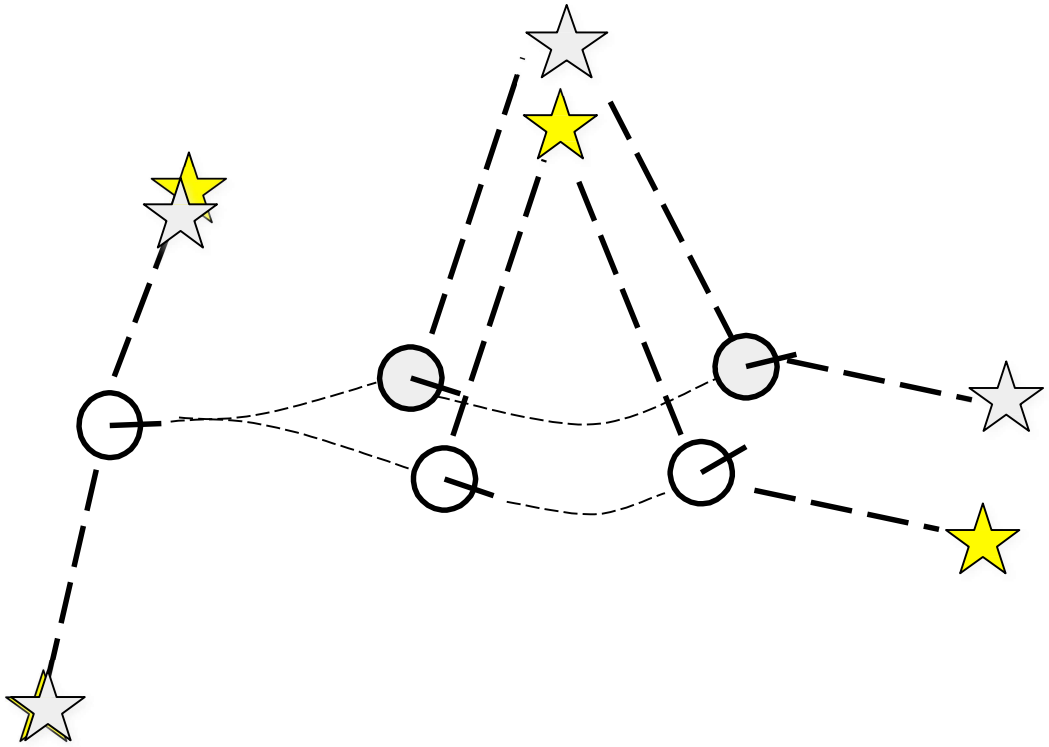
Mapping Example

Estimate the landmarks given the camera's poses



SLAM Example

- Estimate the robot's poses and the landmarks at the same time

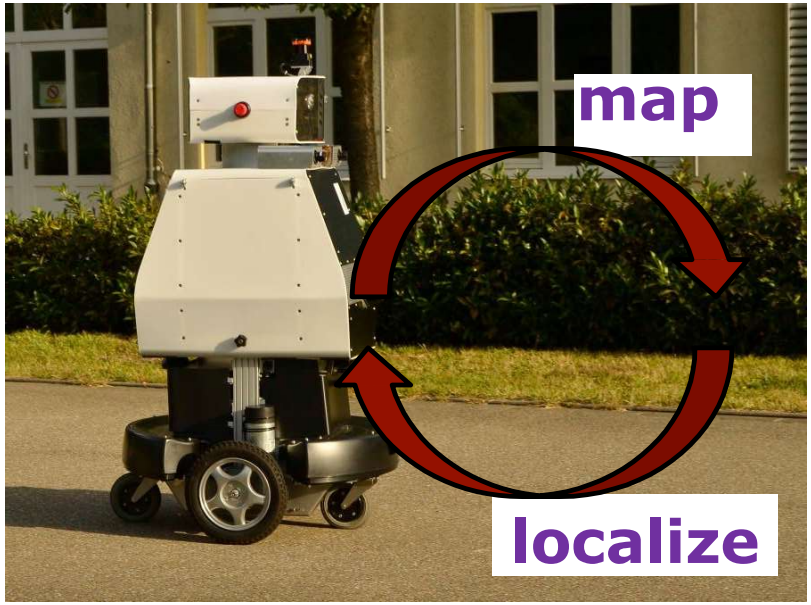


The SLAM Problem

SLAM is a **chicken-or-egg** problem:

→ a map is needed for localization and

→ a pose estimate is needed for mapping



SLAM is Relevant

It is considered a fundamental problem for inside-out tracking

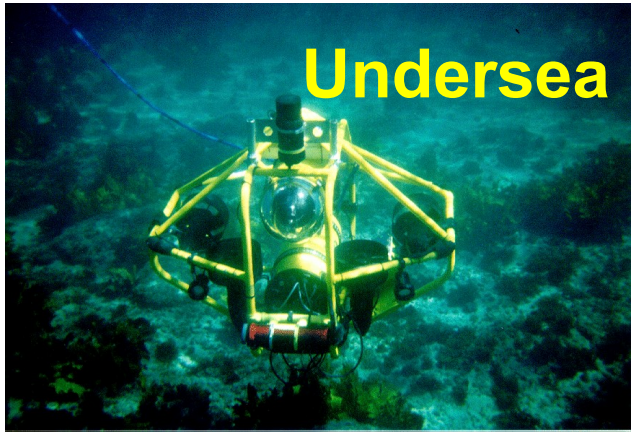
SLAM is the basis for most navigation systems

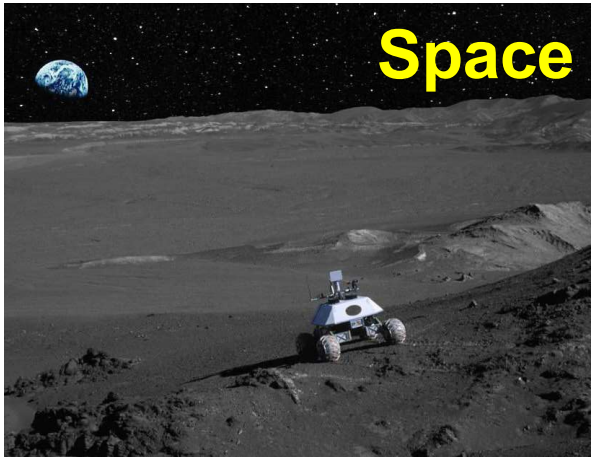
- SLAM is central to a range of indoor, outdoor, air and underwater applications for both manned and autonomous vehicles.

Examples:

- At home: vacuum cleaner, lawn mower
- Air: surveillance with unmanned air vehicles
- Underwater: reef monitoring
- Underground: exploration of mines
- Space: terrain mapping for localization

Additional SLAM Applications





Courtesy of Evolution Robotics, H. Durrant-Whyte, NASA, S. Thrun

Definition of the SLAM Problem

Given

- The camera's controls
- Observations

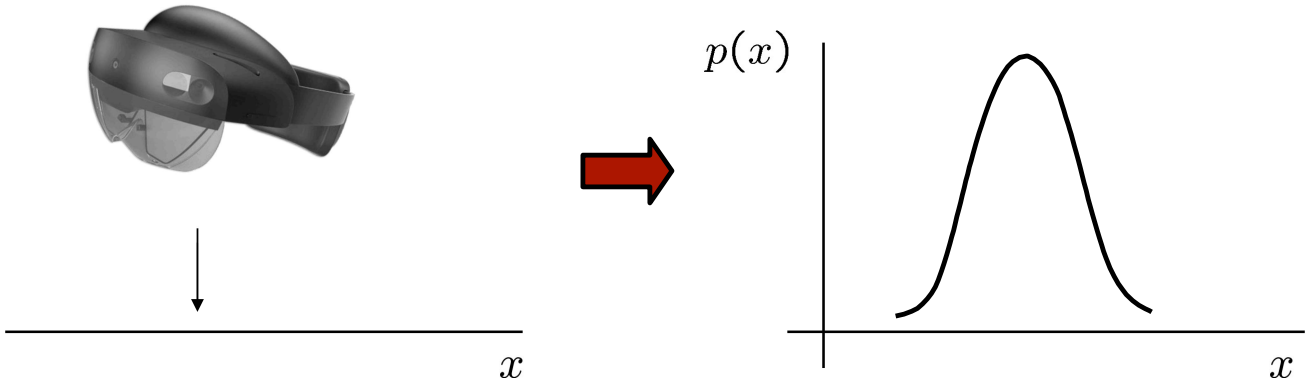
Wanted

- Map of the environment
- Path of the camera

$$x_{0:T} = \{x_0, x_1, x_2 \ldots, x_T\}$$

Probabilistic Approaches

- Uncertainty in the camera's motions and observations
- Use the probability theory to explicitly represent the uncertainty



"The user is exactly here"

"The robot is somewhere here"

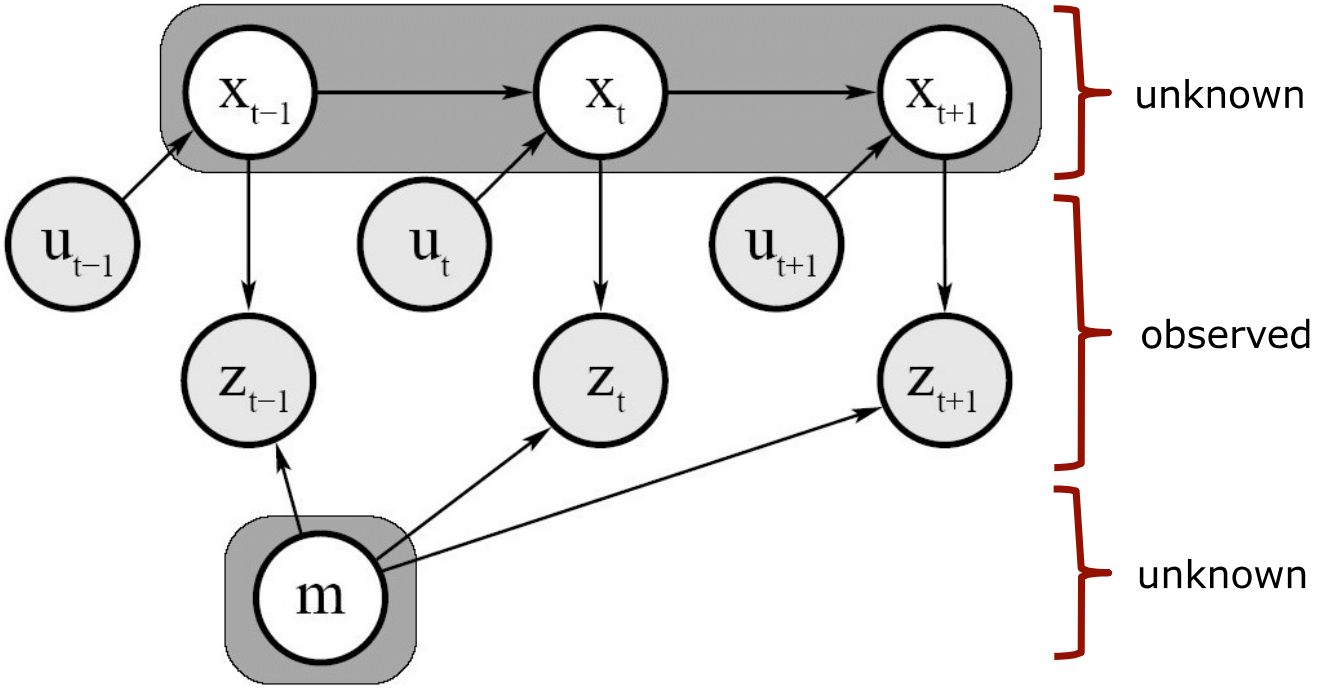
In the Probabilistic World

Estimate the robot's path and the map

$$p(x_{0:T}, m \mid z_{1:T}, u_{1:T})$$

distribution path map given observations controls

Graphical Model



$$p(x_{0:T}, m \mid z_{1:T}, u_{1:T})$$

Full SLAM vs. Online SLAM

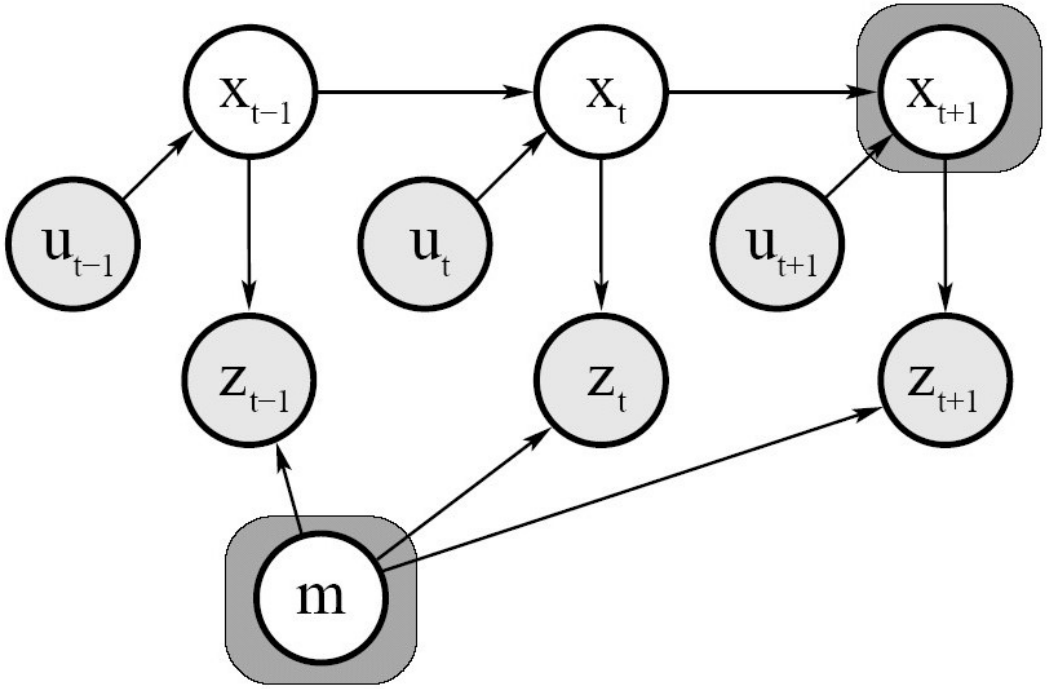
Full SLAM estimates the entire path

$$p(x_{0:T}, m \mid z_{1:T}, u_{1:T})$$

Online SLAM seeks to recover only the most recent pose

$$p(x_t, m \mid z_{1:t}, u_{1:t})$$

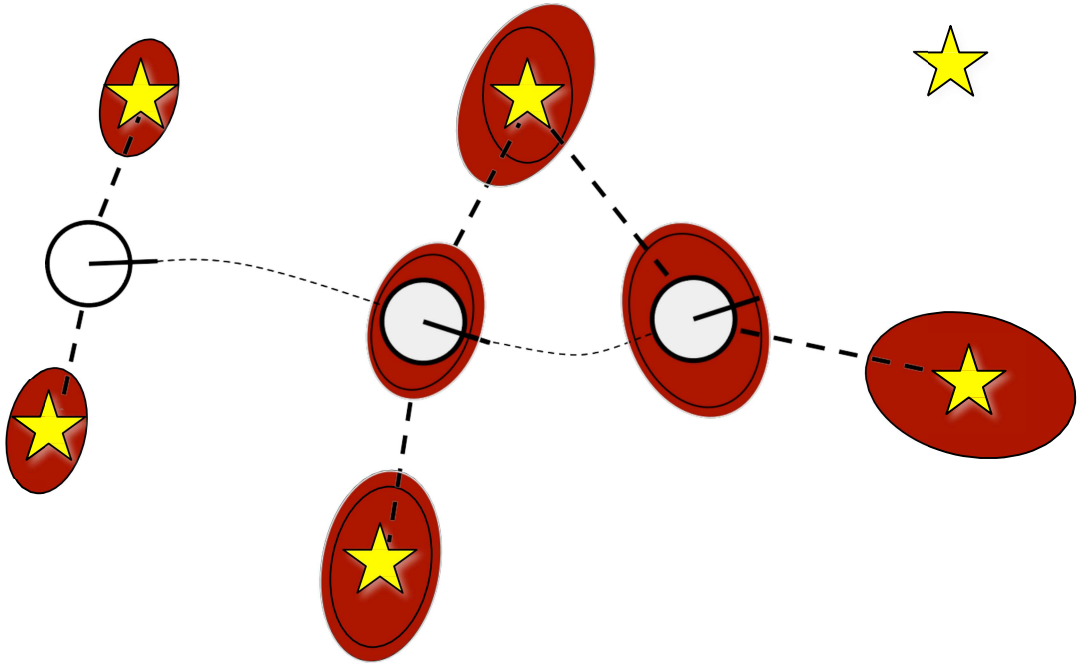
Graphical Model of Online SLAM



$$p(x_{t+1}, m \mid z_{1:t+1}, u_{1:t+1})$$

Why is SLAM a Hard Problem?

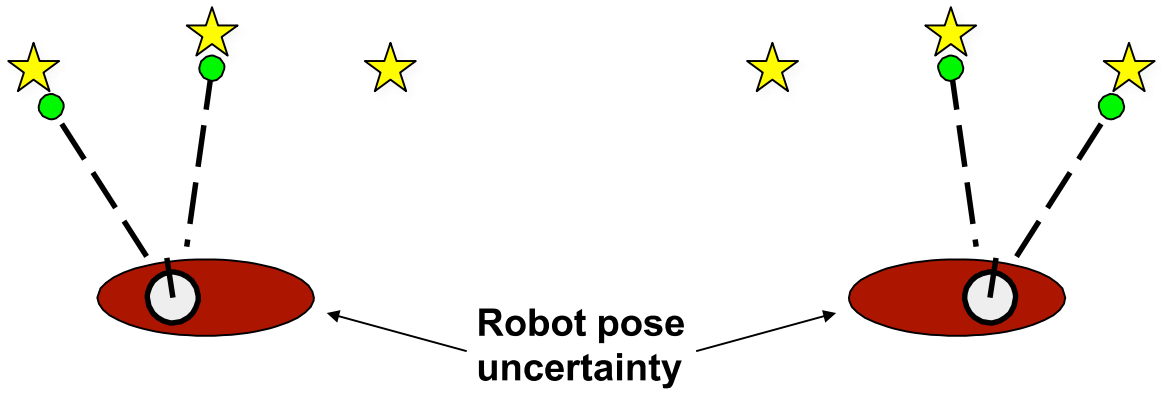
1. Camera path and map are both **unknown**



2. Map and pose estimates correlated

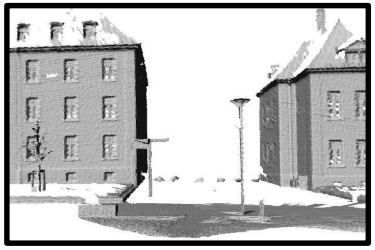
Why is SLAM a Hard Problem?

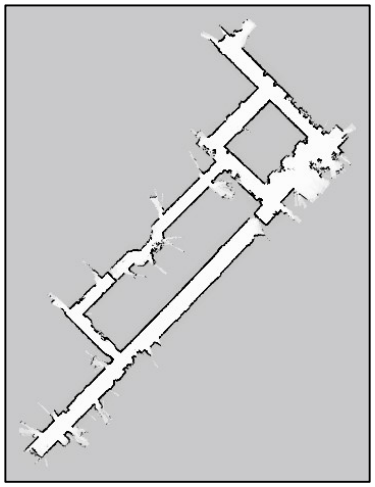
- The **mapping between observations and the map is unknown**
- Picking **wrong** data associations can have **catastrophic** consequences (divergence)

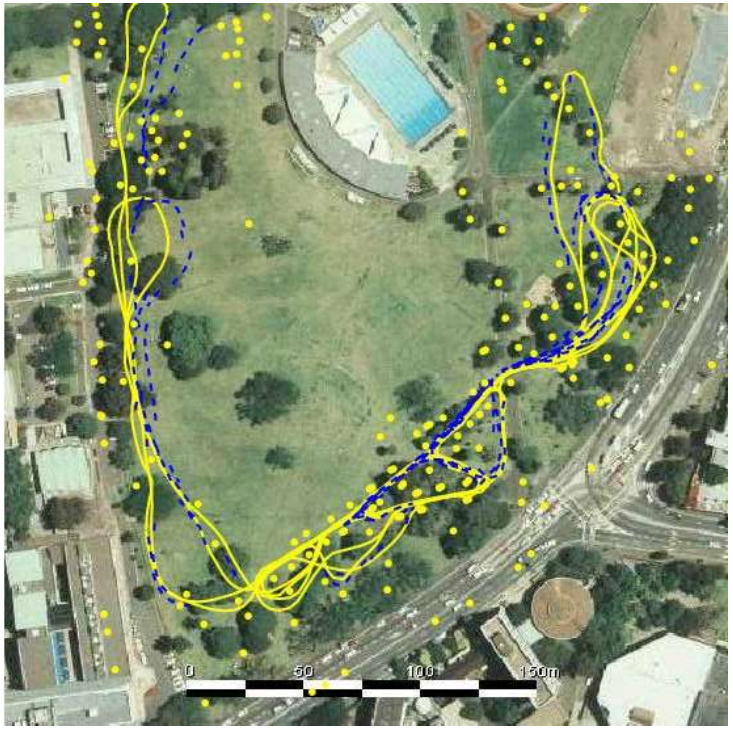


Taxonomy of the SLAM Problem

Volumetric vs. feature-based SLAM



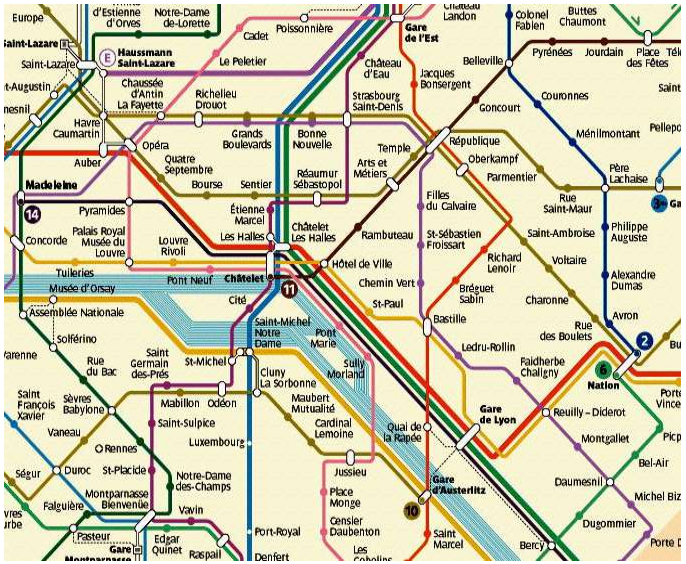




Courtesy by E. Nebot

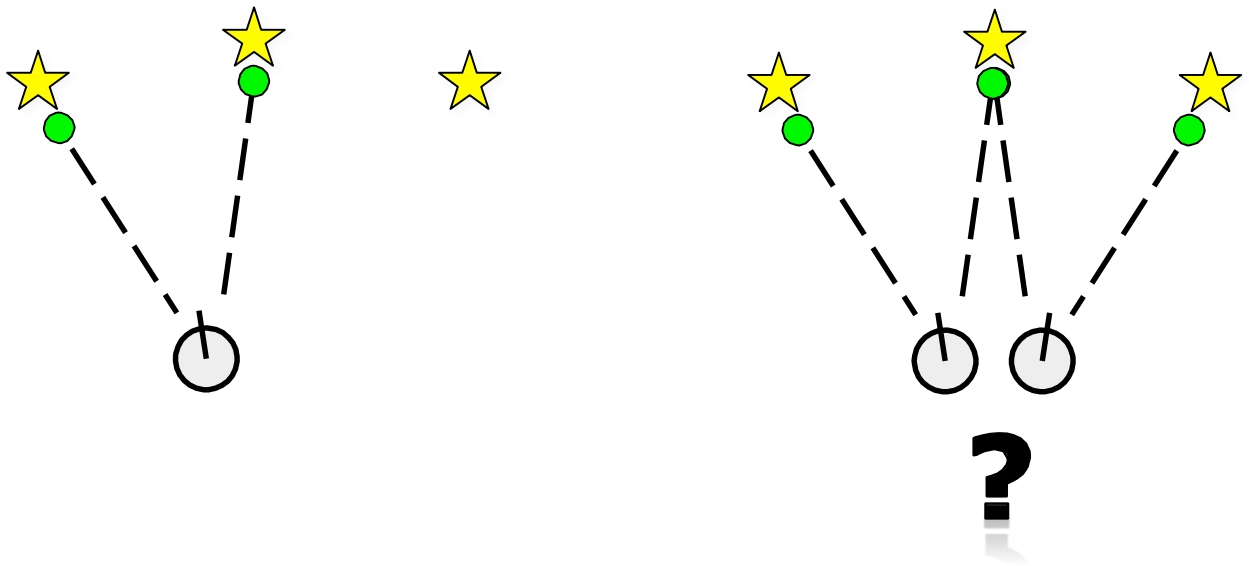
Taxonomy of the SLAM Problem

Topologic vs. geometric maps



Taxonomy of the SLAM Problem

Known vs. unknown correspondence

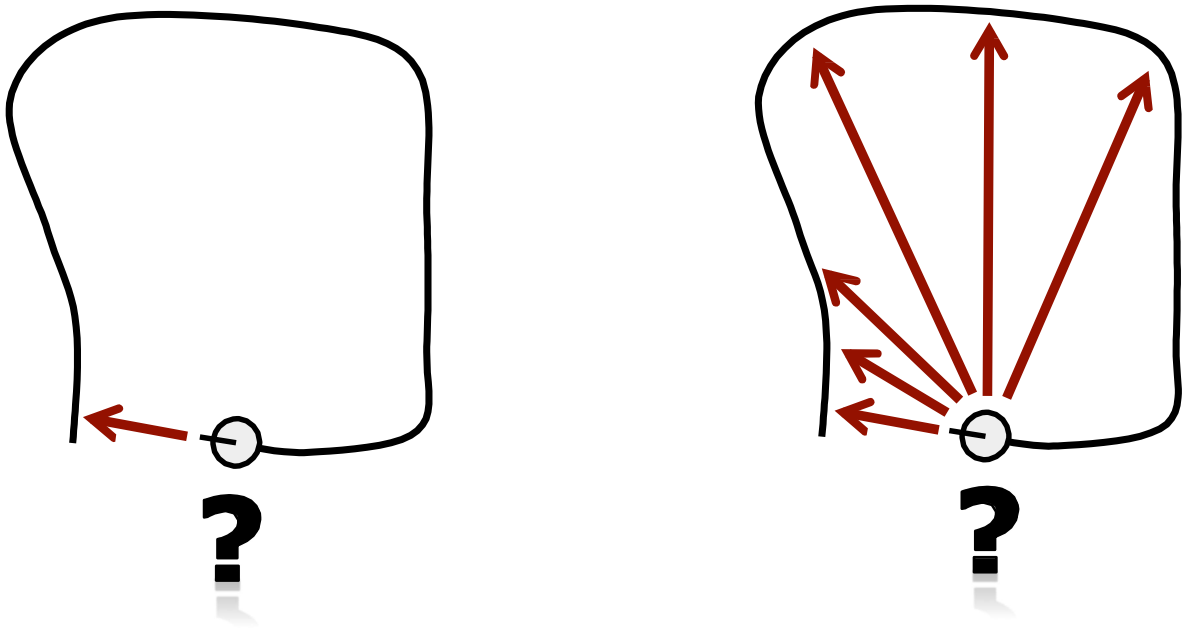


Taxonomy of the SLAM Problem

Static vs. dynamic environments

Taxonomy of the SLAM Problem

Small vs. large uncertainty



Taxonomy of the SLAM Problem

Active vs. passive SLAM

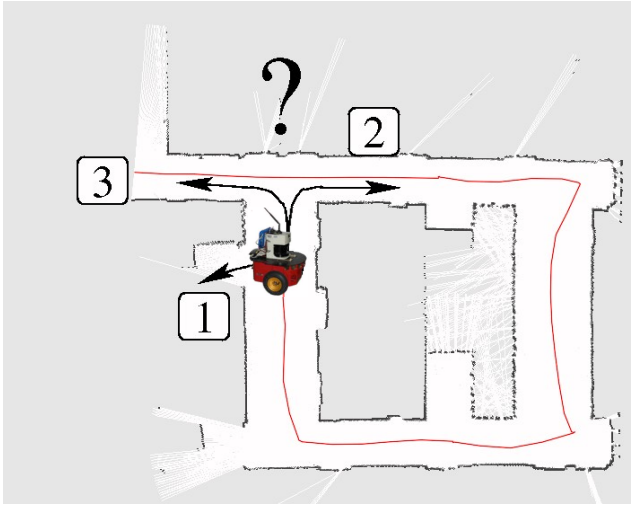


Image courtesy by Petter Duvander

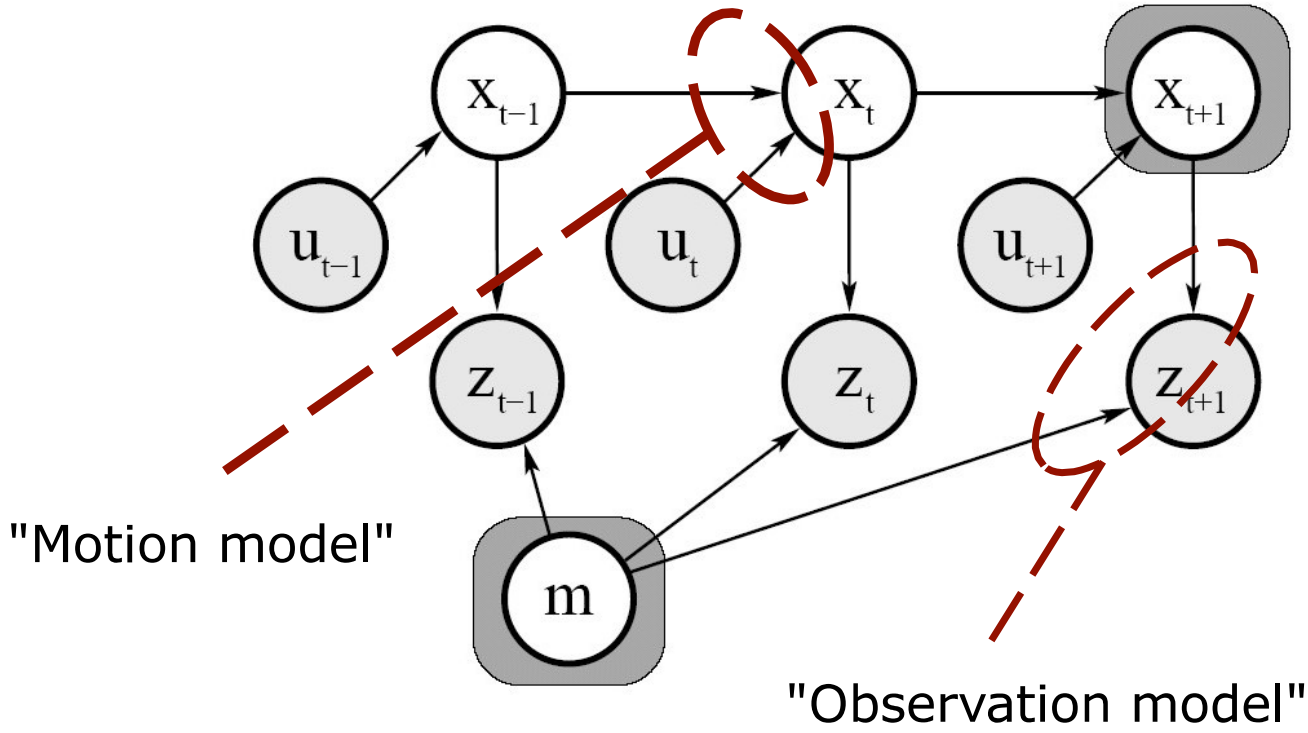
Taxonomy of the SLAM Problem

Single-device vs. multi-device SLAM

Approaches to SLAM

- Large variety of different SLAM approaches have been proposed
- Most robotics conferences dedicate multiple tracks to SLAM
- The majority of techniques uses probabilistic concepts
- History of SLAM dates back to the mid-eighties
- Related problems in geodesy and photogrammetry

Motion and Observation Model



Motion Model

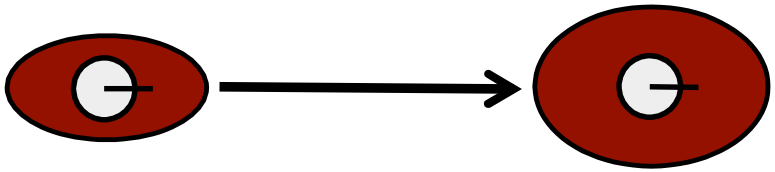
The motion model describes the relative motion of the camera

$$p(x_t \mid x_{t-1}, u_t)$$

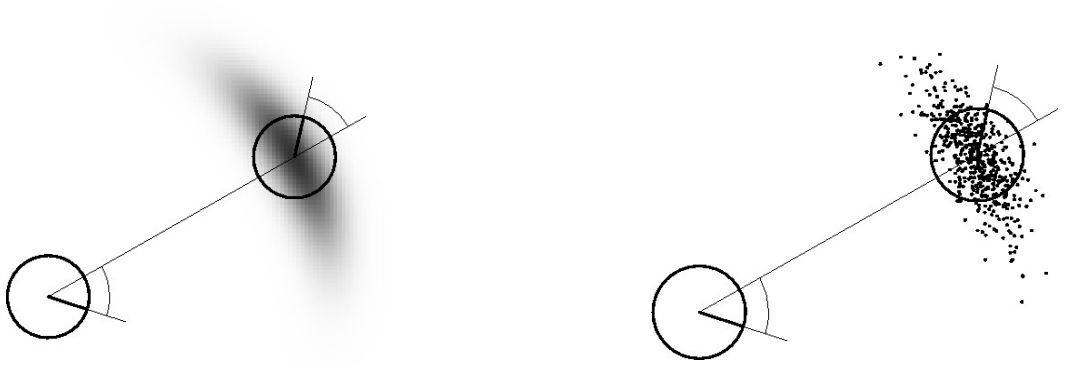
distribution new pose given old pose control

Motion Model Examples

Gaussian model



Non-Gaussian model



Observation Model

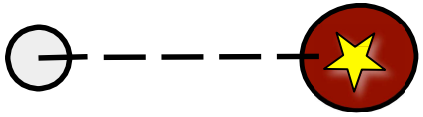
The observation or sensor model relates measurements with the camera's pose

$$p(z_t \mid x_t)$$

distribution observation given pose

Observation Model Examples

Gaussian model



Non-Gaussian model