CAP 4453
Robot Vision

Dr. Gonzalo Vaca-Castaño

gonzalo.vacacastano@ucf.edu

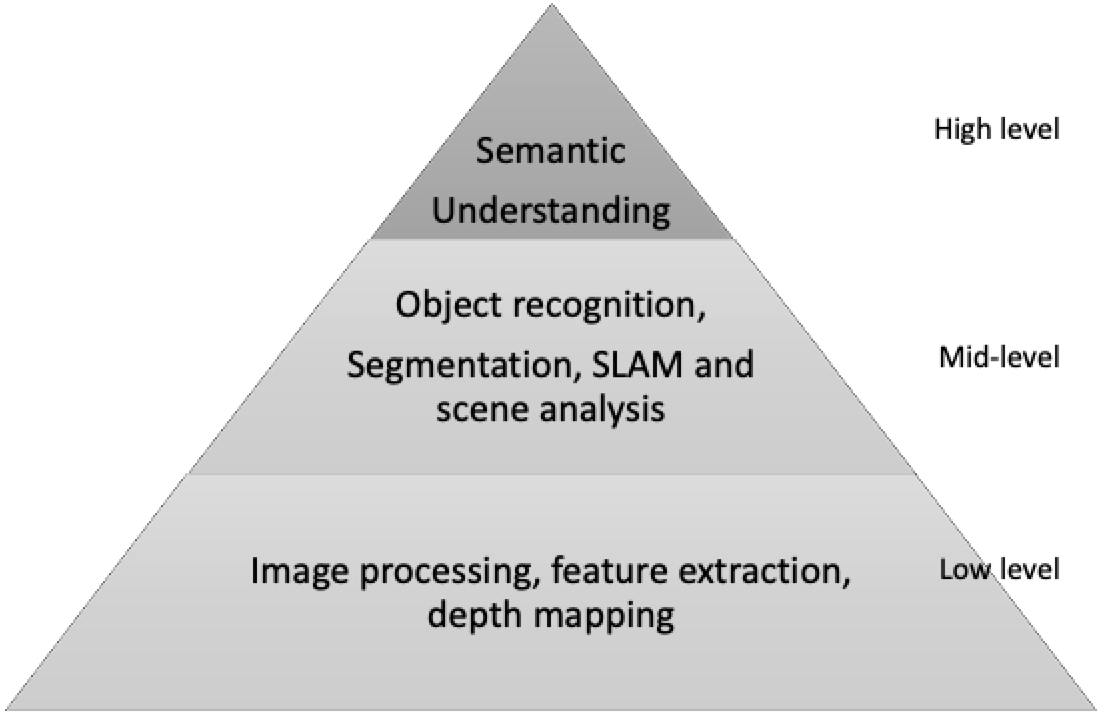
Robot Vision

14. SLAM

Outline

- What is Slam
- Motivation
- Introduction to Visual Slam
- Feature descriptors
  - HOG
  - MOPS
- SIFT

Motivation

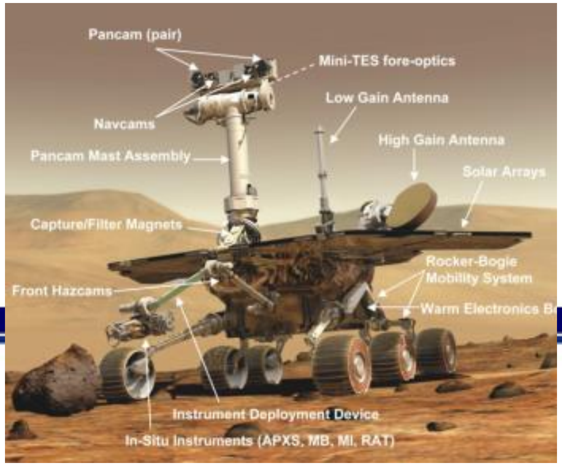

Semantic Understanding
Object recognition, Segmentation, SLAM and scene analysis
Image processing, feature extraction, depth mapping
High level
Mid-level
Low level

What is SLAM?

- Simultaneous Localization And Mapping
- A general problem: A robot with quantitative sensors, navigating in a previously unknown environment, mapping the environment and calculate its ego-motions.
- In simple words: estimate the robot poses, and meanwhile map the scene.

Examples

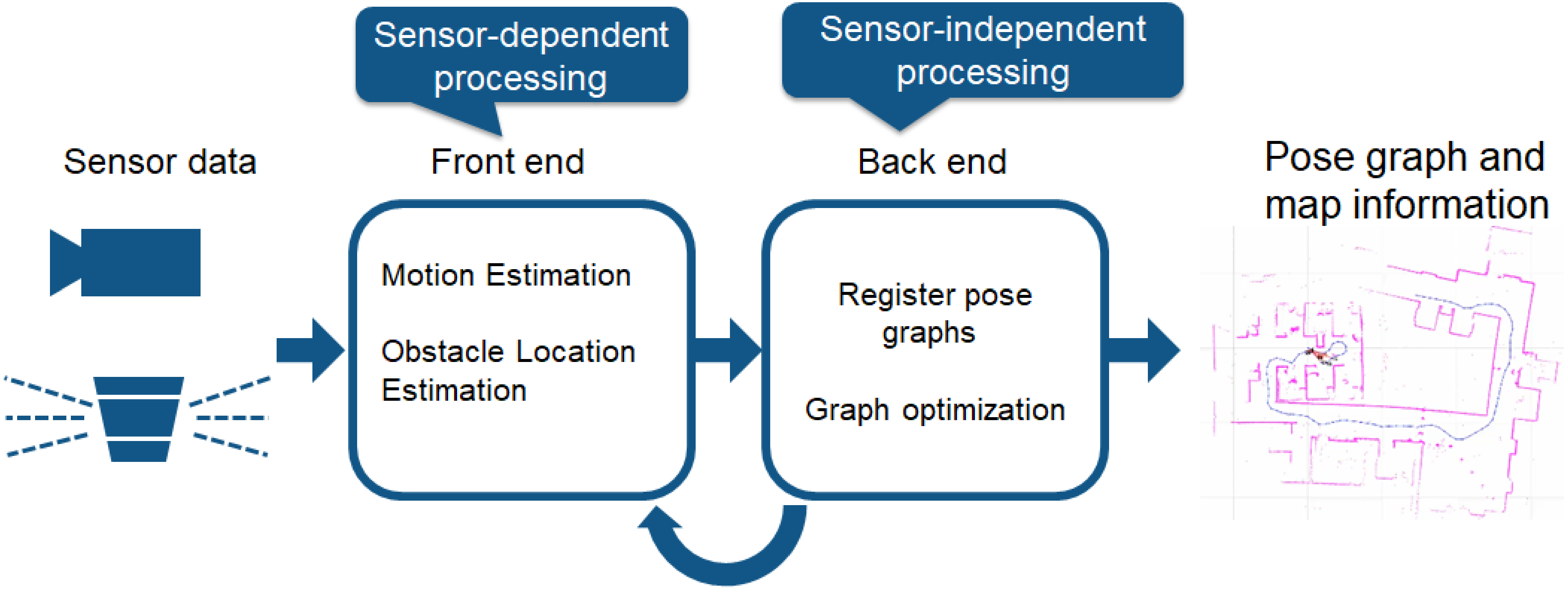
SPOT
CLEAN
DOCK
iRobot
Roomba



Pancam (pair)
Mini-TES fore-optics
Low Gain Antenna
Navcams
High Gain Antenna
Pancam Mast Assembly
Solar Arrays
Capture/Filter Magnets
Rocker-Bogie Mobility System
Front Hazcams
Warm Electronics B
Instrument Deployment Device
In-Situ Instruments (APXS, MB, MI, RAT)

NASA
Jet Propulsion Laboratory
California Institute of Technology
JPL
SCENARIO
Altitude
2000m
1700m
1400m
1st image
2nd image
3rd image
Stephan Weiss
Computer Vision Group
NASA-JPL / CalTech
Stephan.Weiss@ieee.org
First Pair Tracking
Second Pair Tracking
MER-A/Spirit, Gusev Crater, January 4th, 2004
(c) 2013 California Institute of Technology. Government sponsorship acknowledged.

SLAM

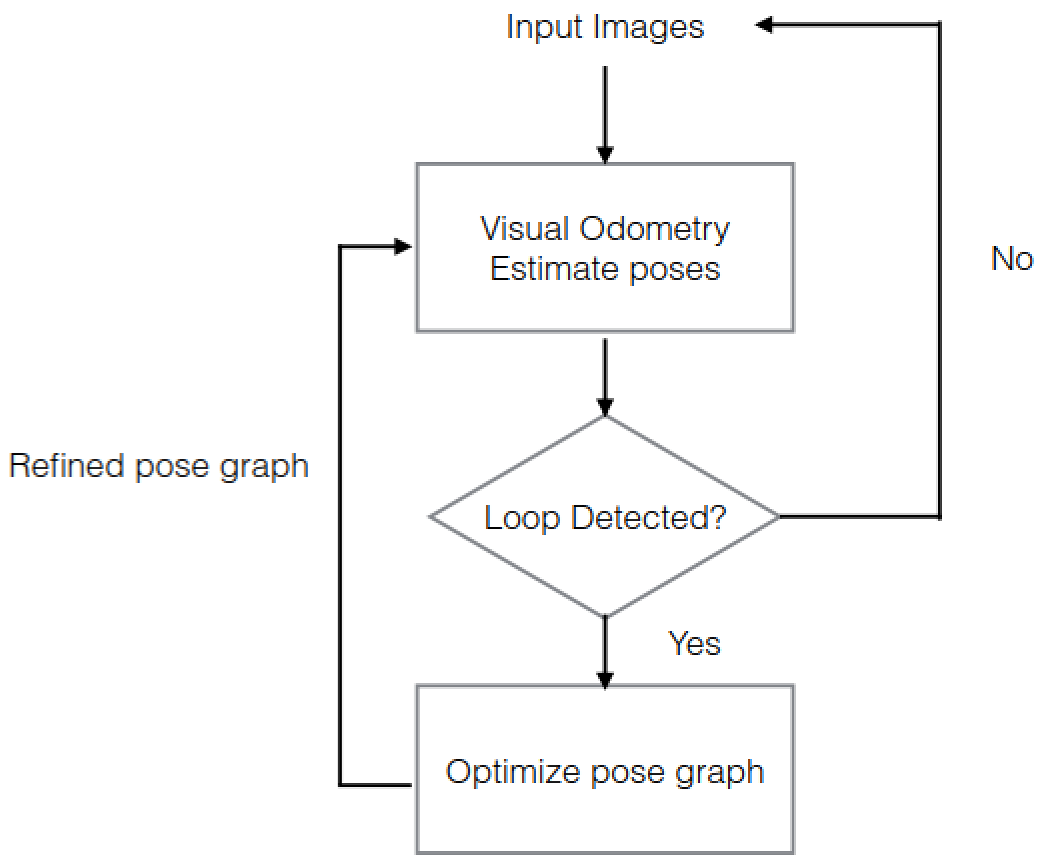


Sensor-dependent processing
Sensor-independent processing
Sensor data
Front end
Back end
Pose graph and map information
Motion Estimation
Obstacle Location Estimation
Register pose graphs
Graph optimization

Multiple versions of Slam (Front-end)

- **Visual SLAM**
  - **Monocular (Orb_SLAM, LSD_SLAM, DSO, etc)**
  - Stereo (ORB_SLAM2, OV2SLAM, …)
  - RGBD (DVO_SLAM , PlanarSLAM, badslam, RESLAM ,…)
- Visual Inertial SLAM (inertial sensors)
  - Monocular (OKVIS, ROVIO, LARVIO, …)
  - Stereo (msckf_vio, Basalt, ICE-BA , …)
  - RGBD
- Lidar Based SLAM
  - LOAM_Livox, FAST-LIO, ….

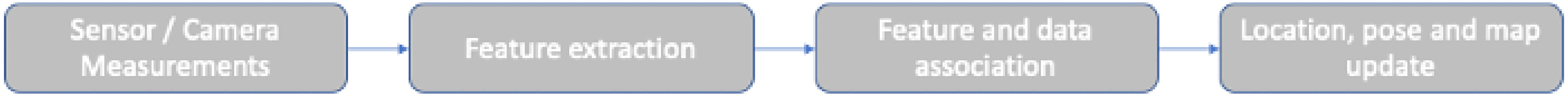
General Visual Slam pipeline



Visual Odometry

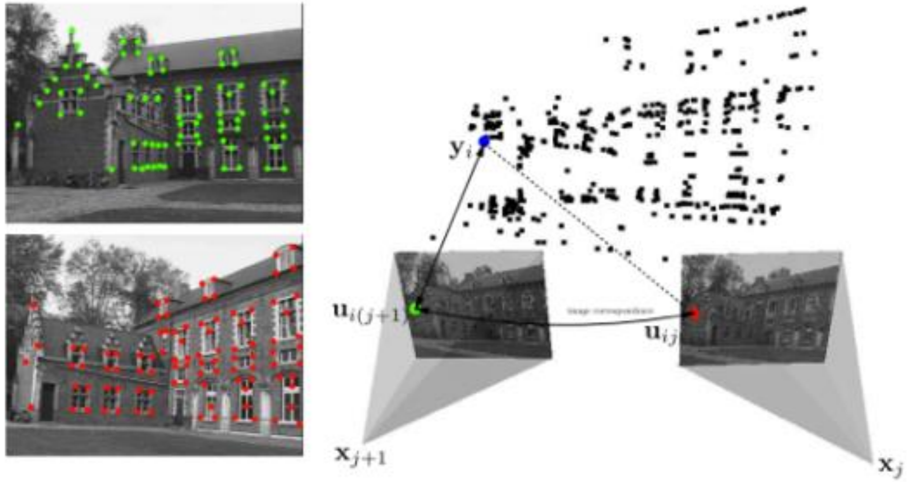
- Goal: estimate the camera movement between adjacent frames (ego-motion) and generate a rough local map.
- We want to estimate 6-DoF camera pose [R|T] incrementally

Visual odometry (monocular)



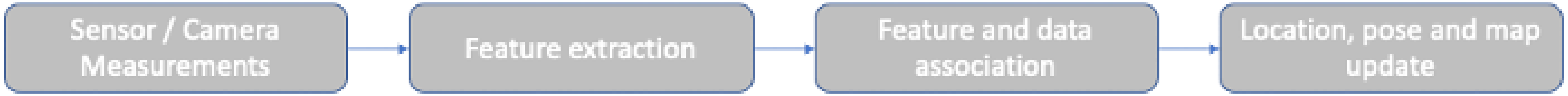
Simplified generic feature-based SLAM process

- Triangulation can be done in consecutive frames



[M.Pollefeys, Hand-held acquisition of 3D models with a video camera., 1999, 3DIM]

Visual odometry (monocular)



Simplified generic feature-based SLAM process

- Feature Extraction: Feature points

  - Detect local features in each image
    - SIFT gives good results (can also use SURF, Harris, etc.)

Visual odometry (monocular)

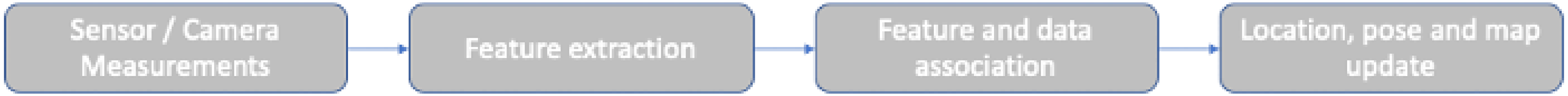
Sensor / Camera Measurements → Feature extraction → Feature and data association → Location, pose and map update

Simplified generic feature-based SLAM process

- Feature and Data Association
  - Tracked features (optical flow)
  - Use Ransac for temporal association

- Temporally match features between frame t and t-1

Visual odometry (monocular)



Simplified generic feature-based SLAM process

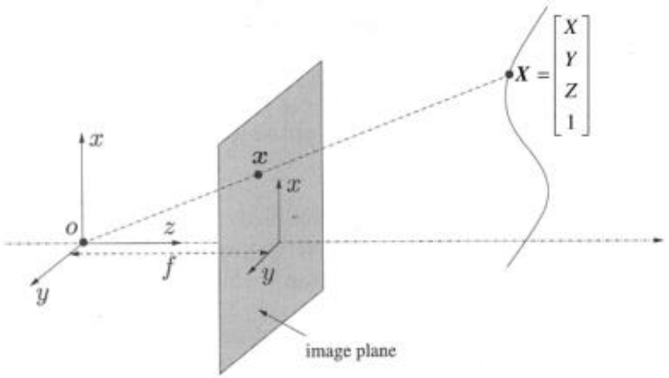
- Camera pose estimation

  3D point transformation: $\mathbf{X}_2 = R\mathbf{X}_1 + T$

  - Estimate R, T from epipolar geometry

- Linear 8-point algorithm $Ae=0$
  - Problem is only of dimension 5 (3 for rotation, 2 for translation up to scale)
  - Linear formulation is fast and simple to solve

- Non-linear 5-point algorithm (Nistér PAMI 204)
  - Finding roots of cubic polynomials
  - Mathematically hairy but fast implementations exist



Theorem of intersecting lines:

$$x = f\frac{X}{Z}, \quad y = f\frac{Y}{Z} \quad \text{or} \quad \boldsymbol{x} = \begin{bmatrix}x\\y\end{bmatrix} = \frac{f}{Z}\begin{bmatrix}X\\Y\end{bmatrix}$$

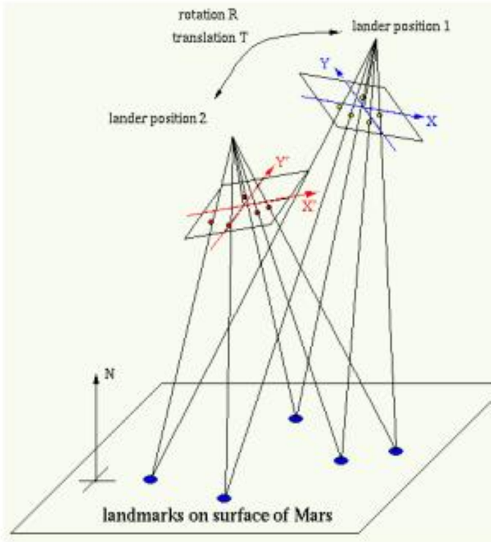
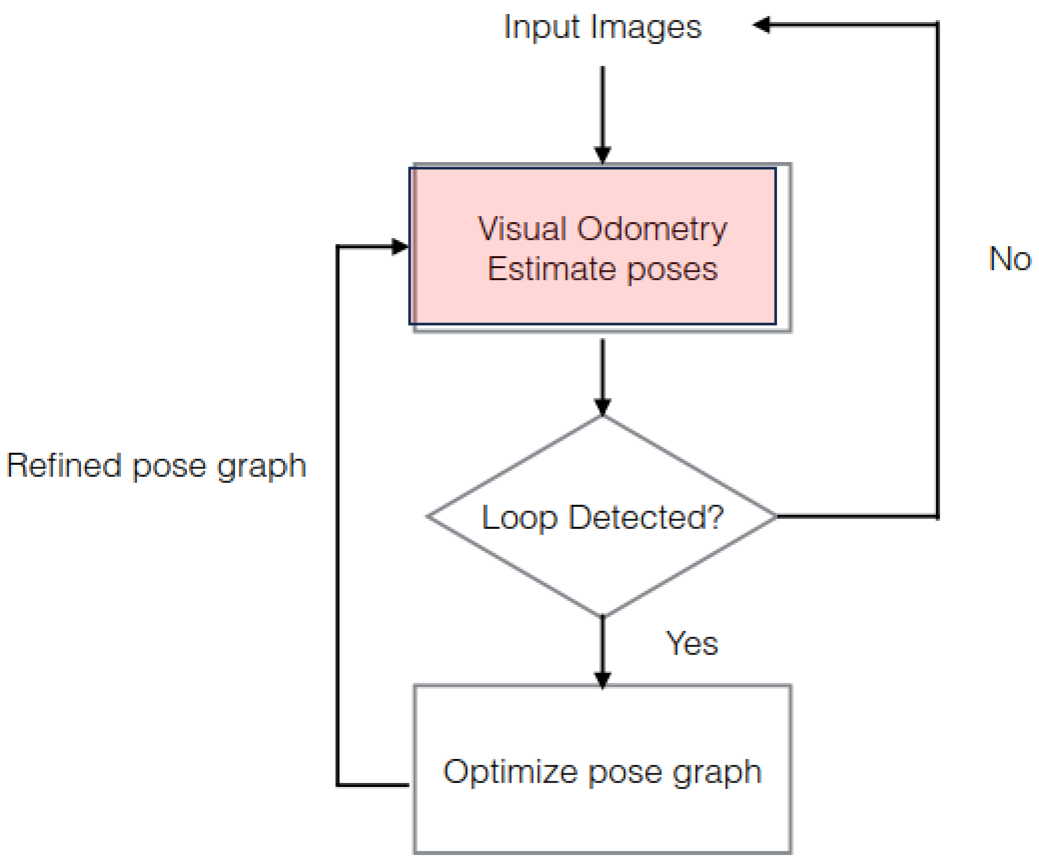


Image: Ma, Y., Soatto, S., Kosecká, J., Sastry, S.S. : "An Invitation to 3D Vision"

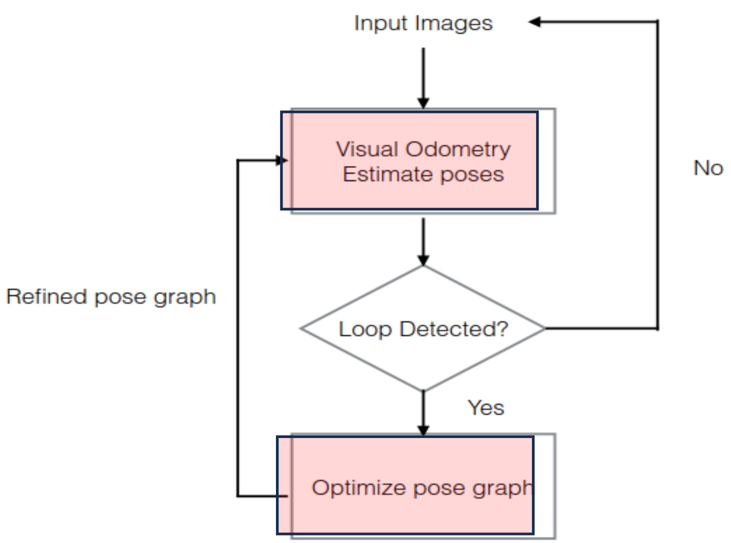
assume calibrated camera

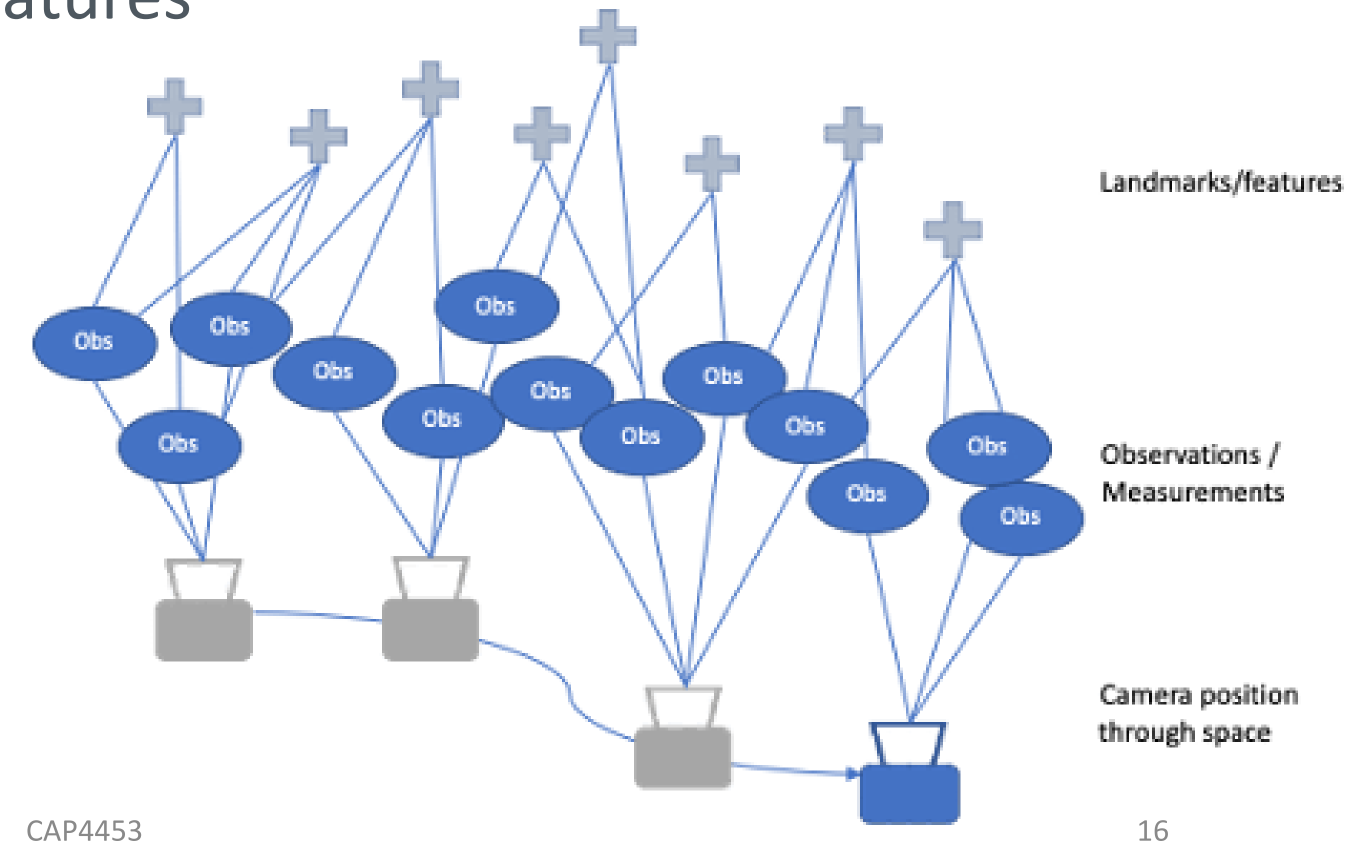
Are we done ?


Input Images
Visual Odometry
Estimate poses
No
Refined pose graph
Loop Detected?
Yes
Optimize pose graph

Optimize pose graph (backend)

- Features → location → map of features (relative to cameras)
- New measurements →prune features

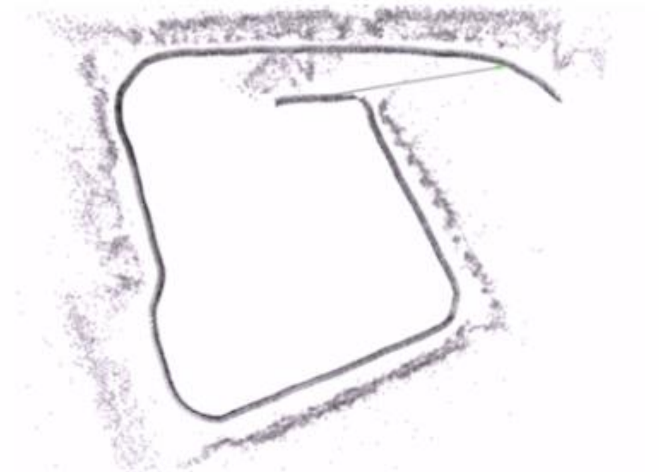


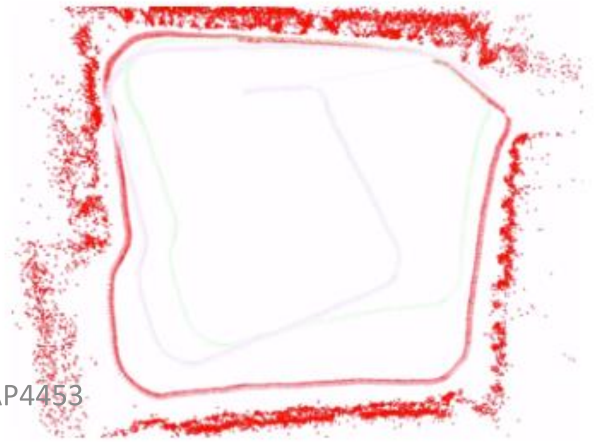


Features and observations from the camera through space

Optimize pose graph (backend)

- As the camera moves through space, there is increasing noise and uncertainty between the images the camera captures and its associated motion.
- backend optimization (optimization pose graph) mainly refers to the process of dealing with the noise in SLAM systems
- We only have numbers and matrices in the backend (state estimation) without those beautiful images (computer vision)

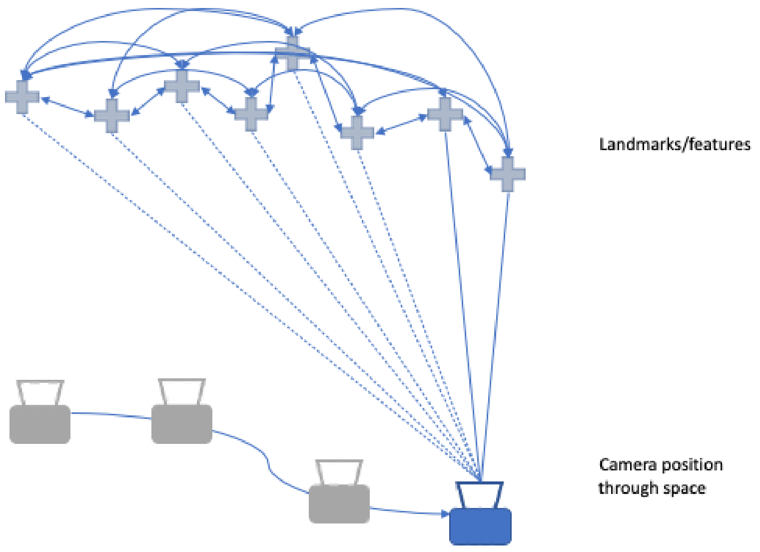




Optimize pose graph (back end) Approaches

Extended Kalman Filters(EKF)

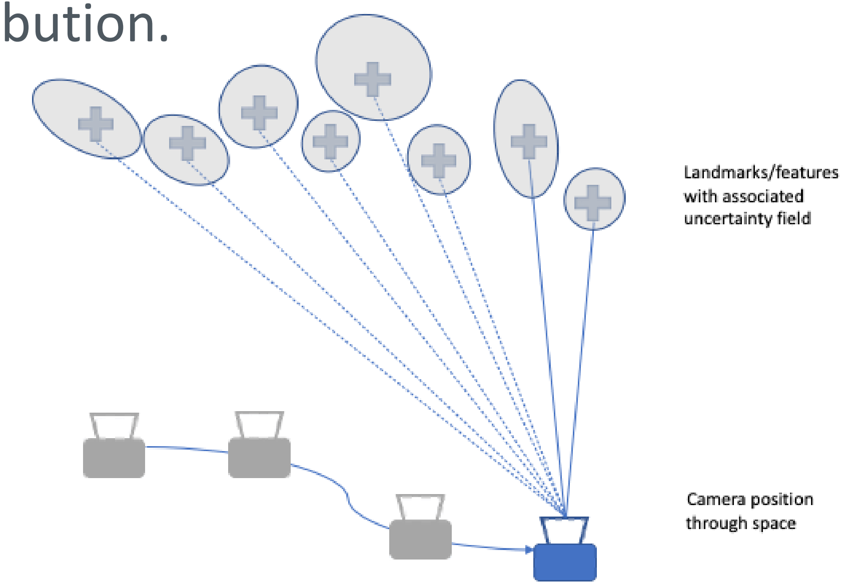
- takes nonlinear systems, and linearizes the predictions and measurements around their mean.



After Kalman filtering

Particle filters

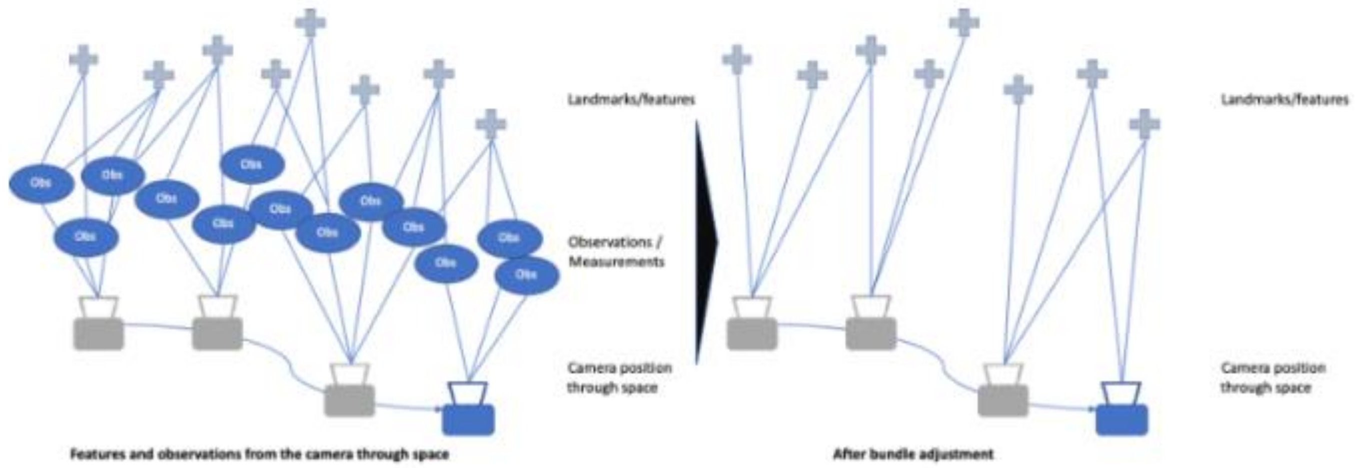
- Each feature point as a particle
- At each measurement uncertainty is updated against the predicted position.
- Unlike Kalman filters, particle filters can handle noise from any distribution, and states can have a multi-modal distribution.



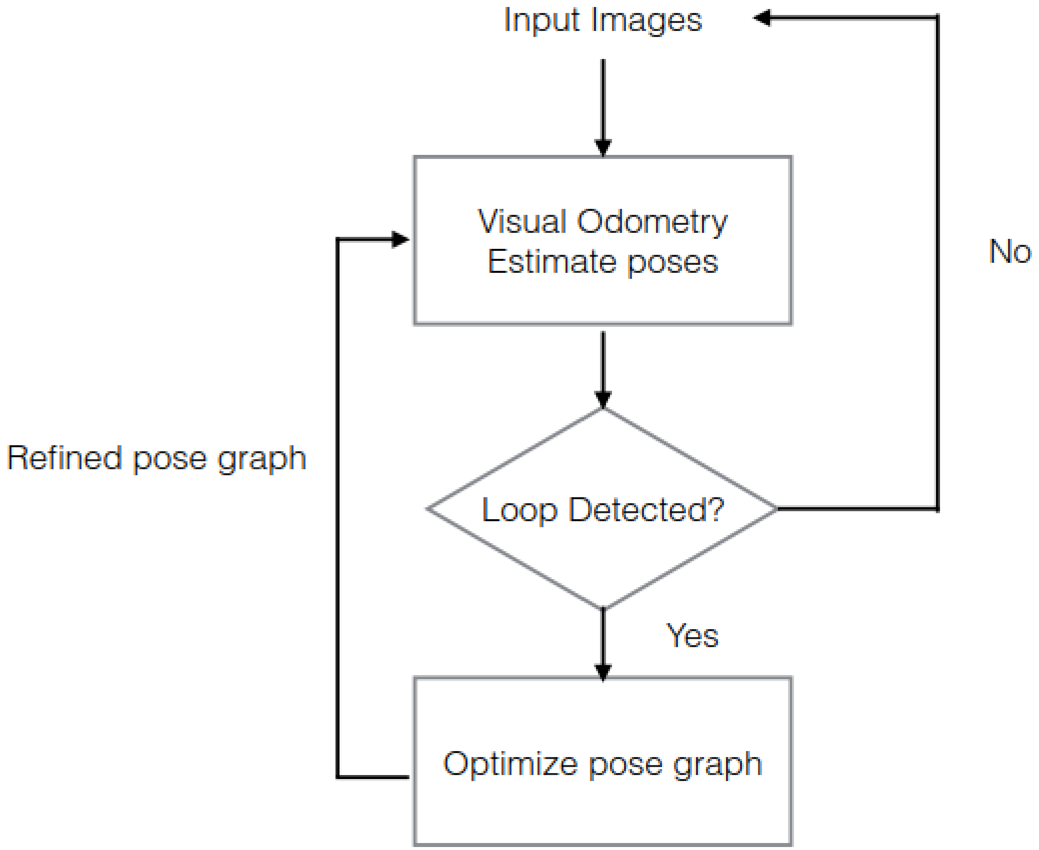
After Particle filtering

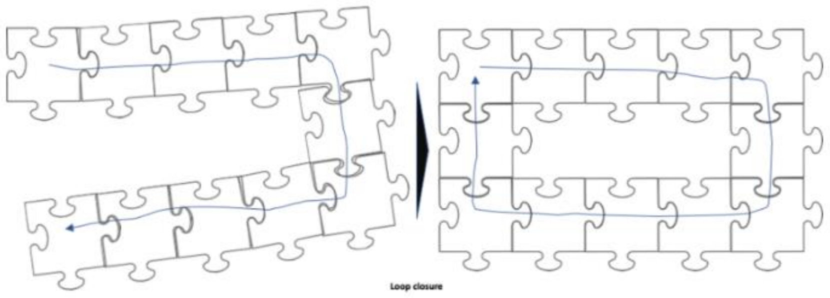
Bundle adjustment

- It is a batch operation, and not performed on every captured frame.
- onlinear least squares operation on the current model. Imagine a “bundle” of light rays from all the features connected to each of the camera observations, and “adjusted” to optimize these connections directly to the sensor position and orientation as in the figure below.



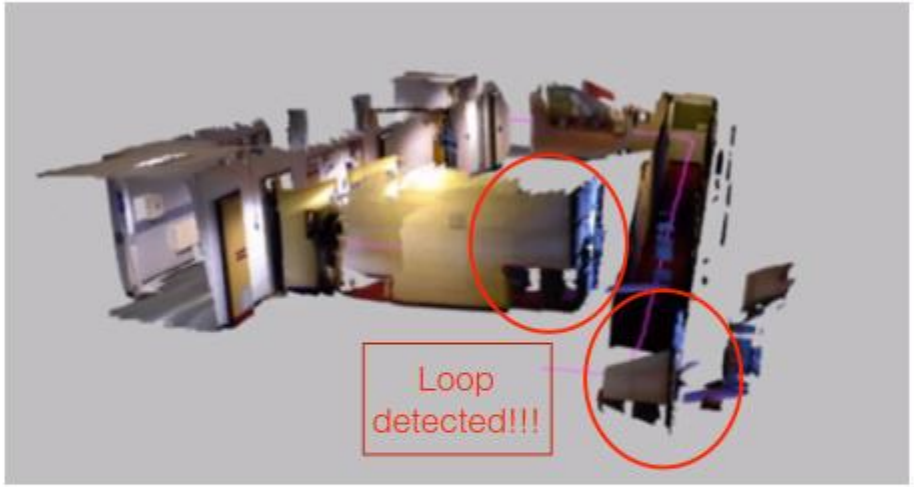
Loop closure

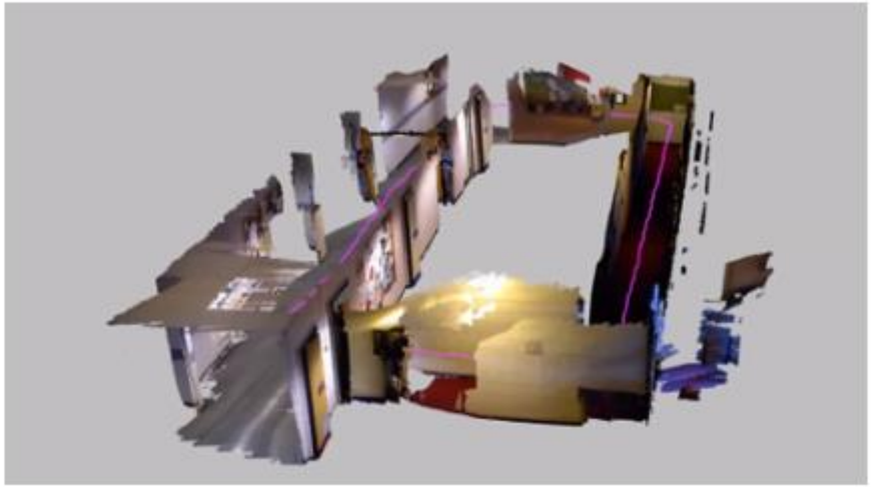




What's in between — Loop Detection

- Optimization works the best if we have global correspondence from landmarks.
- When a loop is detected, we can set up the global correspondence.





[T.Whelan et al. Deformation-based Loop Closure for Large Scale Dense RGB-D SLAM, 2013, IROS]

Resources

- Book: https://github.com/gaoxiang12/slambook-en
- CVPR 2014 tutorial: Visual SLAM Tutorial | at CVPR 2014, June 28 (room C 213-215) (cmu.edu)
- Links to everything in SLAM: Awesome-SLAM | A curated list of SLAM resources (silenceoverflow.github.io)

Questions?