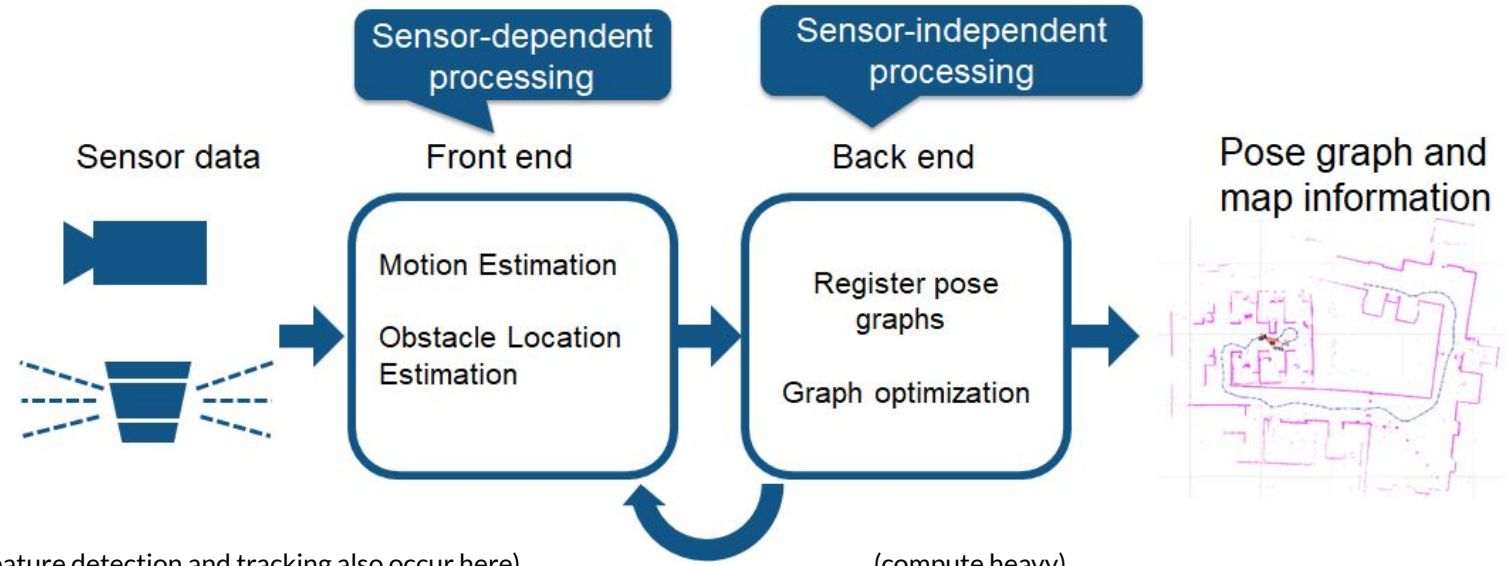
SLAM modules

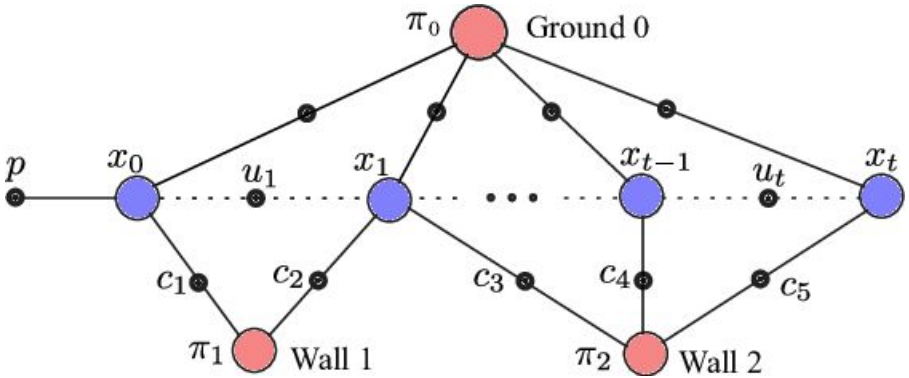
Anatomy of SLAM systems



(feature detection and tracking also occur here)

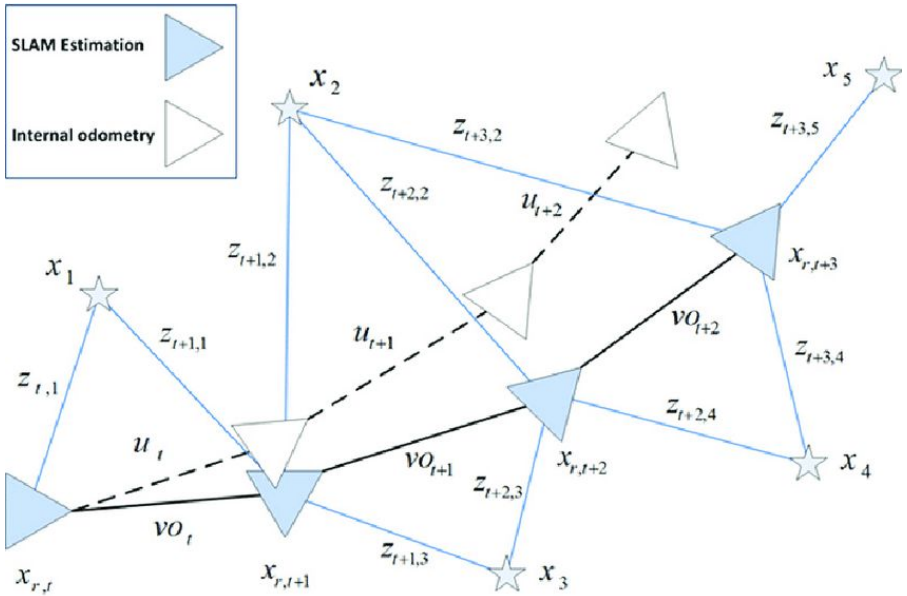
(compute heavy)

Pose graphs and graph based formulations



- Encode observations, poses, odometry etc. as nodes in your pose graphs
- Edges between two nodes represent constraints to optimise over
  - Odometry constraints
  - Loops
  - Landmark constraints

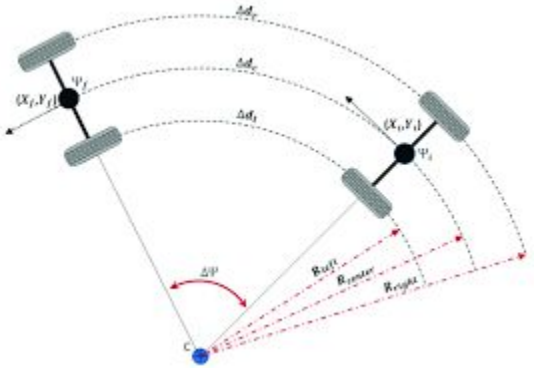
Pose graphs graph based formulations



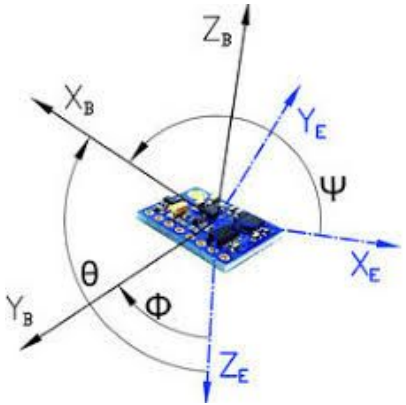
- Use this pose graph to calculate a total ‘loss’
- The loss tells us how closely our estimate of the robots poses and our mapping satisfies each constraint
- Compensate for uncertainty in one reading with information from other readings
- Formulate a large mathematical model for this loss using all constraints
- Use a non-linear solver to optimise over this loss

Getting constraints (odometry)

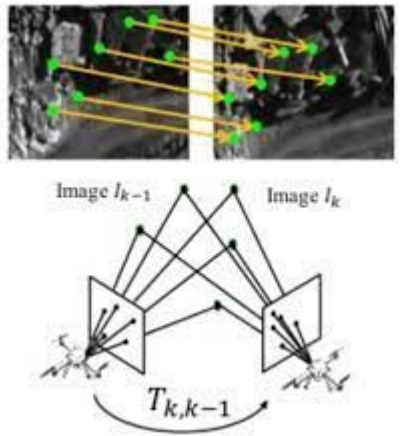
- Feedback from sensors onboard our robot give us valuable constraints between sequential poses



Wheel odometry



IMU odometry



Visual odometry

Getting constraints (landmarks)

- Use a feature detector and tracker to keep track of “significant” features the robot encounters
- If the same feature is spotted again, you have a landmark constraint
- The feature must be visible in both poses regardless of what our noisy readings say
- Use the features position across both images to create a virtual constraint.
-