B31YS Robotics Systems Science

# Week 7

# Visual Odometry and SLAM

Dr. Sen Wang

s.wang@hw.ac.uk

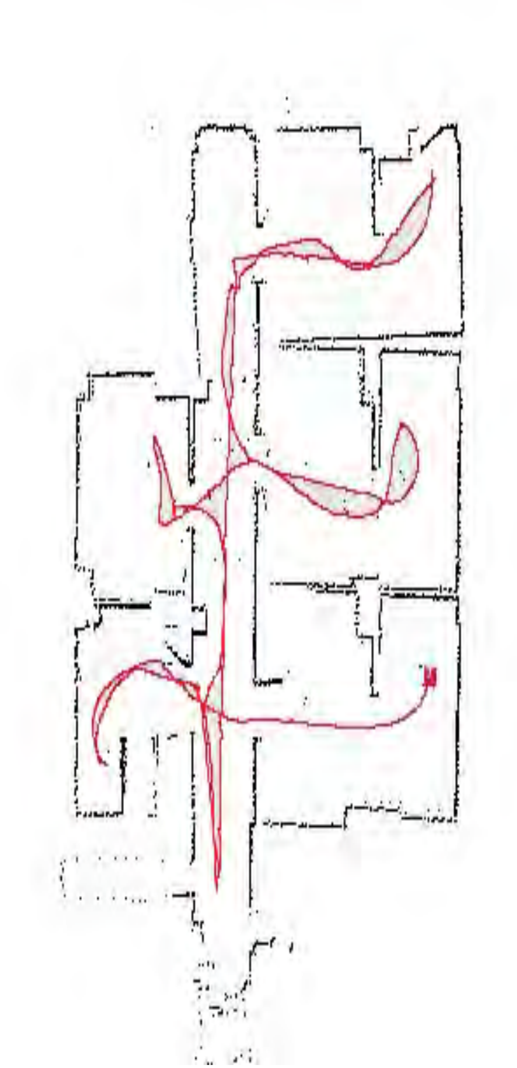
slide courtesy: Burgard, Stachniss, Bennewitz, Tipaldi, Spinello, Tipaldi, etc.

# Schedule this Week

| | Monday | | | Thursday | Friday |
|---|---|---|---|---|---|
| | 10.15 – 11.15 | 13.15 – 15.15 | 16.15 – 17.15 | 16.15 – 17.15 | 11.15 – 13.15 |
| Week 7 | Lecture<br>- SLAM formulation<br>- GraphSLAM<br>- Front-End<br>- Back-End<br>- Visual Odometry (sparse method) | Lab – VO/SLAM<br>- read papers on Vision to understand visual odometry/SLAM and choose one to demo on Friday<br>- 3 mins presentation to introduce the algorithm you choose | Lecture<br>- Loop Closure Detection and Visual BoW<br>- Visual Odometry (direct method)<br>-VO and visual SLAM | Lecture<br>- Learning based Methods<br>- Review of KF/ EKF/PF/SLAM | Lab – VO/SLAM<br>- 3 mins presentation and demo of an open-source visual SLAM algorithm on ROS (HARD deadline!) |

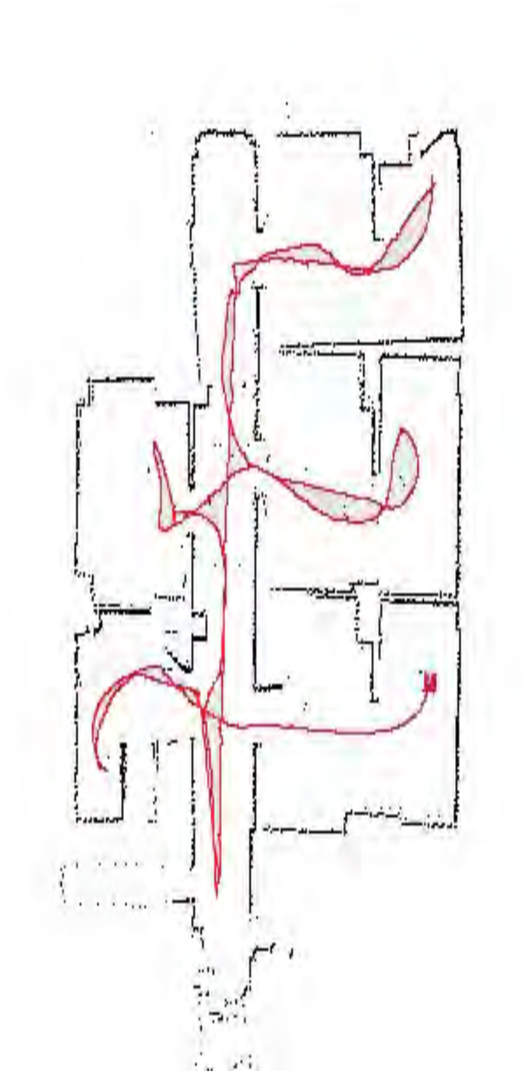
# Simultaneous Localization and Mapping

- Estimate the pose of a robot and the map of the environment **at the same time**
- SLAM is hard because robot path and map are **both unknown**



# Simultaneous Localization and Mapping

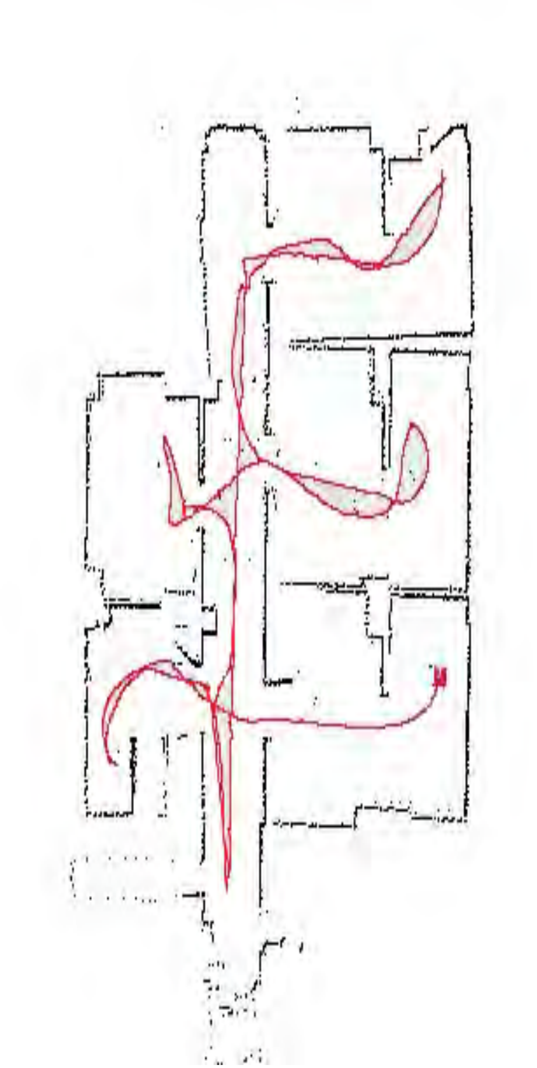
- Estimate the pose of a robot and the map of the environment **at the same time**
- SLAM is hard because robot path and map are **both unknown**



# Simultaneous Localization and Mapping

- Estimate the pose of a robot and the map of the environment **at the same time**
- SLAM is hard because robot path and map are **both unknown**

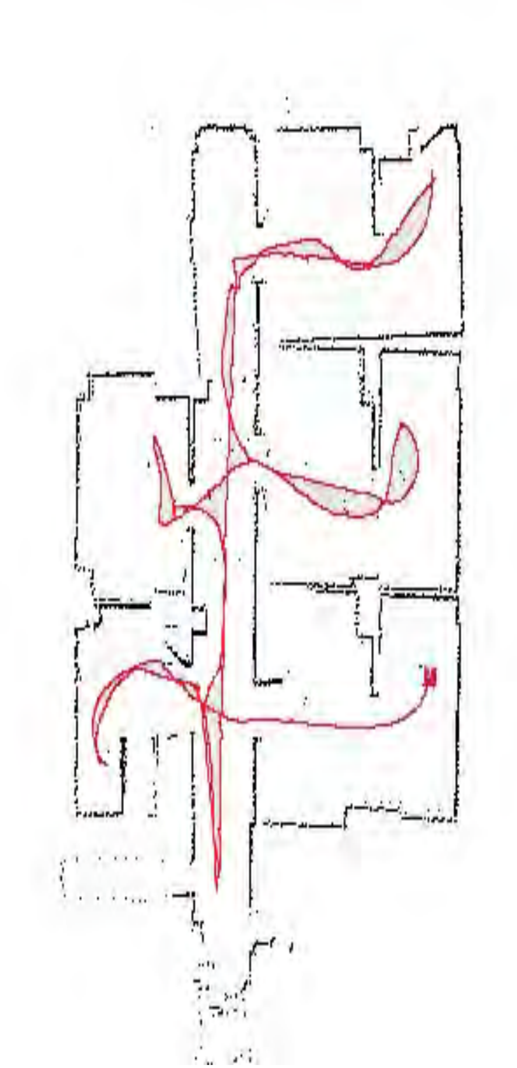
- Localization: inferring location given a map
- Mapping: inferring a map given locations
- SLAM: learning a map and locating the robot simultaneously



# Simultaneous Localization and Mapping

- Estimate the pose of a robot and the map of the environment **at the same time**
- SLAM is hard because robot path and map are **both unknown**

- Localization: inferring location given a map
- Mapping: inferring a map given locations
- SLAM: learning a map and locating the robot simultaneously



# SLAM Problem

- SLAM is a chicken-or-egg problem:
  - a map is needed for localization and
  - a pose estimate is needed for mapping

$$\underbrace{p}_{\text{distribution}}(\underbrace{x_{0:T}}_{\text{path}}, \underbrace{m}_{\text{map}} \underbrace{\mid}_{\text{given}} \underbrace{z_{1:T}}_{\text{observations}}, \underbrace{u_{1:T}}_{\text{controls}})$$

**Given**

- The robot's controls
  $u_{1:T} = \{u_1, u_2, u_3, \ldots, u_T\}$
- Observations
  $z_{1:T} = \{z_1, z_2, z_3, \ldots, z_T\}$

**Wanted**

- Map of the environment
  $m$
- Path of the robot
  $x_{0:T} = \{x_0, x_1, x_2, \ldots, x_T\}$

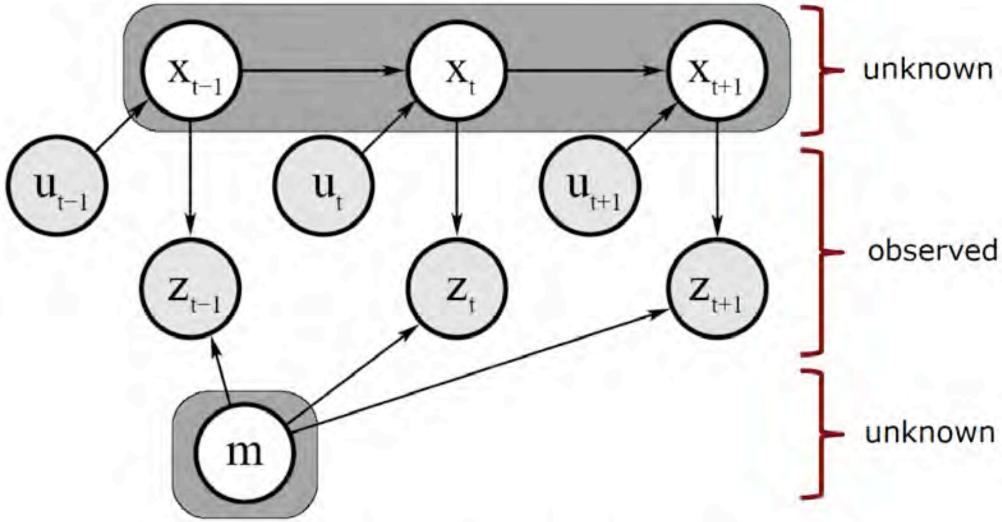
SLAM is the basis for most navigation systems and is considered as a fundamental problem for robots to become truly autonomous

# Full SLAM vs. Online SLAM - Graphical Model

- Full SLAM estimates the entire path

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

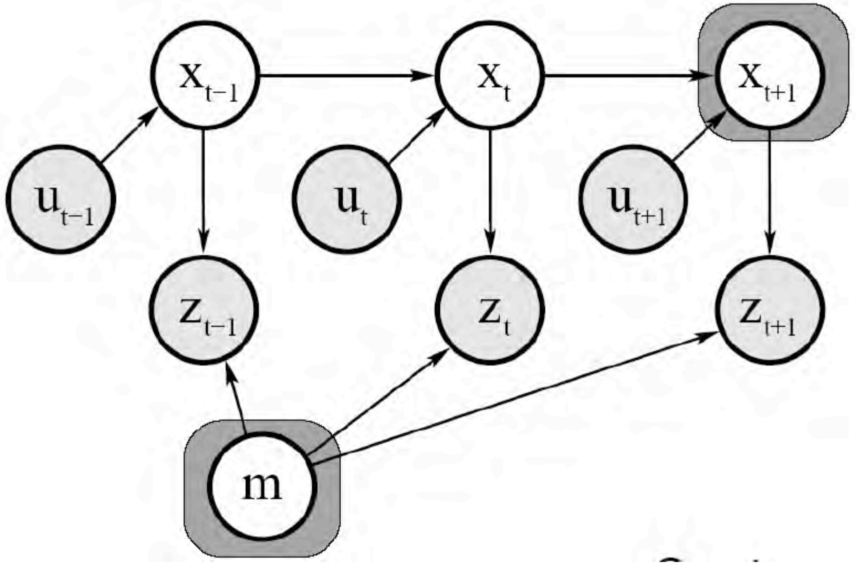
Estimates entire path and map!




- Online SLAM seeks to recover only the most recent pose

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

Estimates most recent pose and map!

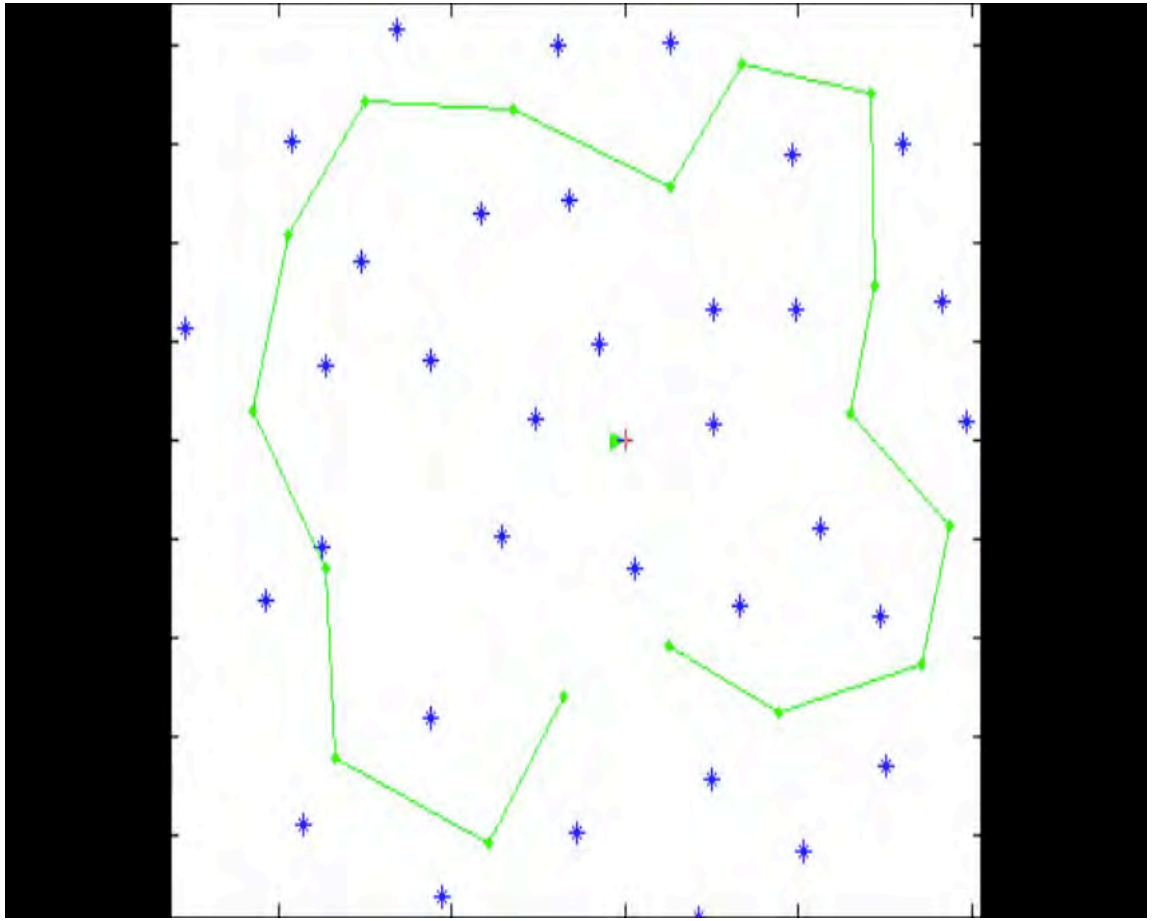


# Three Main Paradigms

| Kalman Filter | Particle Filter | Graph based |
| --- | --- | --- |
| **EKF SLAM** | **FastSLAM** | **GraphSLAM (focus)** |

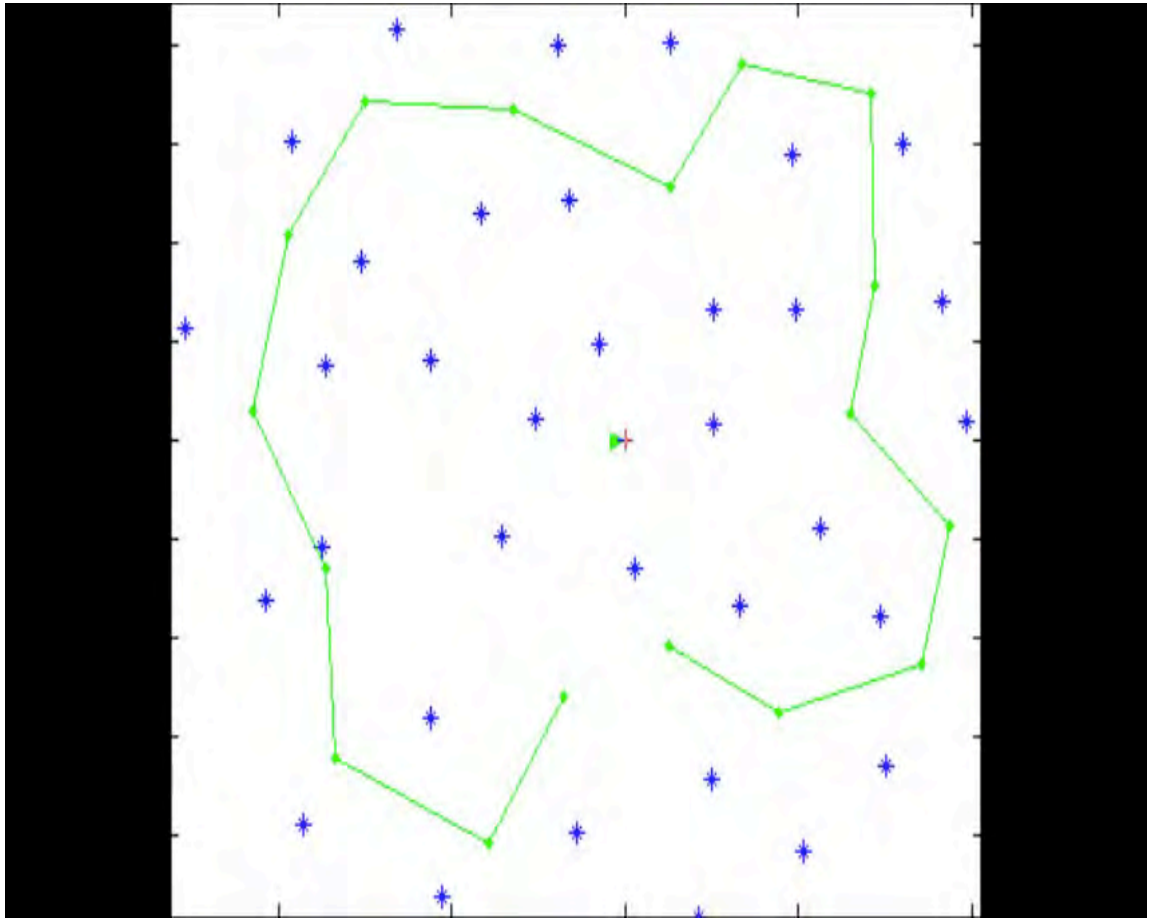
$\underbrace{\text{Kalman Filter EKF SLAM} \quad \text{Particle Filter FastSLAM}}_{\text{Online SLAM}}$ $\underbrace{\text{Graph based GraphSLAM (focus)}}_{\text{Full SLAM}}$

# EKF SLAM



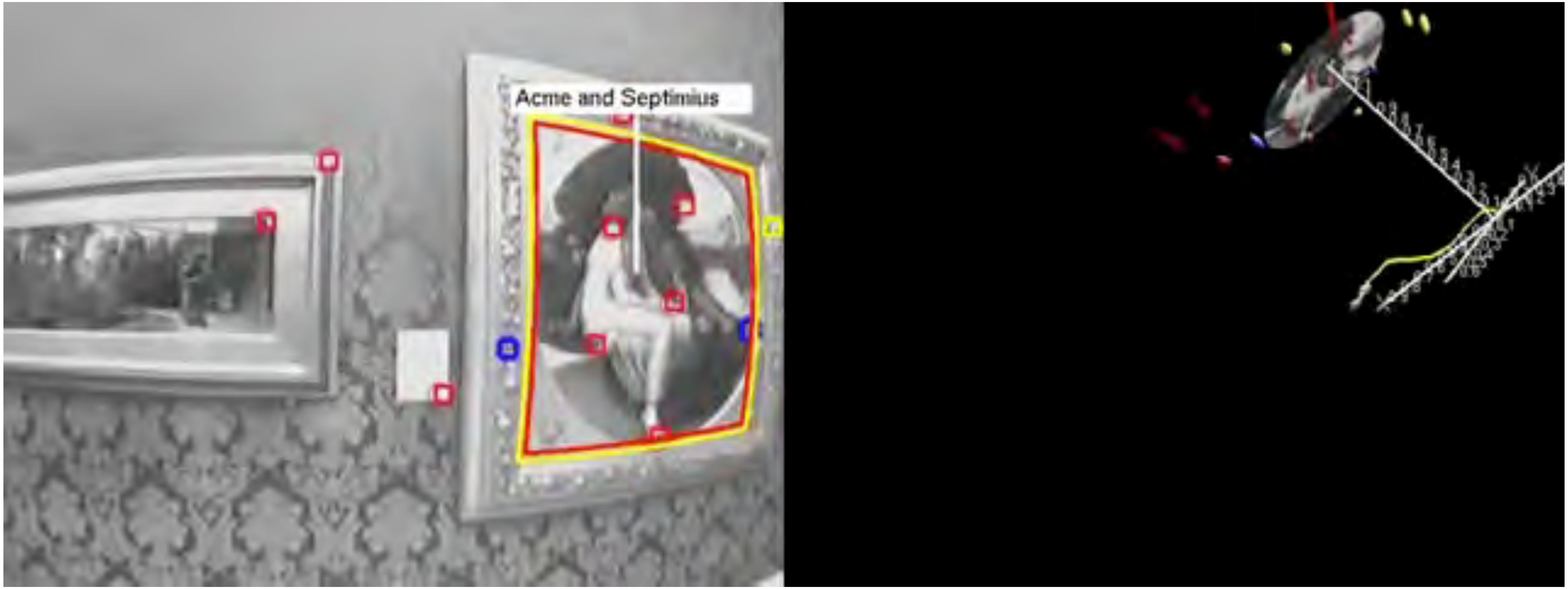
- Sebastian Thrun, Wolfram Burgard, Dieter Fox. Probabilistic robotics.
- Davison, Andrew J., et al. "MonoSLAM: Real-time single camera SLAM." *IEEE Transactions on Pattern Analysis & Machine Intelligence* 6 (2007): 1052-1067.

# EKF SLAM



- Sebastian Thrun, Wolfram Burgard, Dieter Fox. Probabilistic robotics.
- Davison, Andrew J., et al. "MonoSLAM: Real-time single camera SLAM." *IEEE Transactions on Pattern Analysis & Machine Intelligence* 6 (2007): 1052-1067.

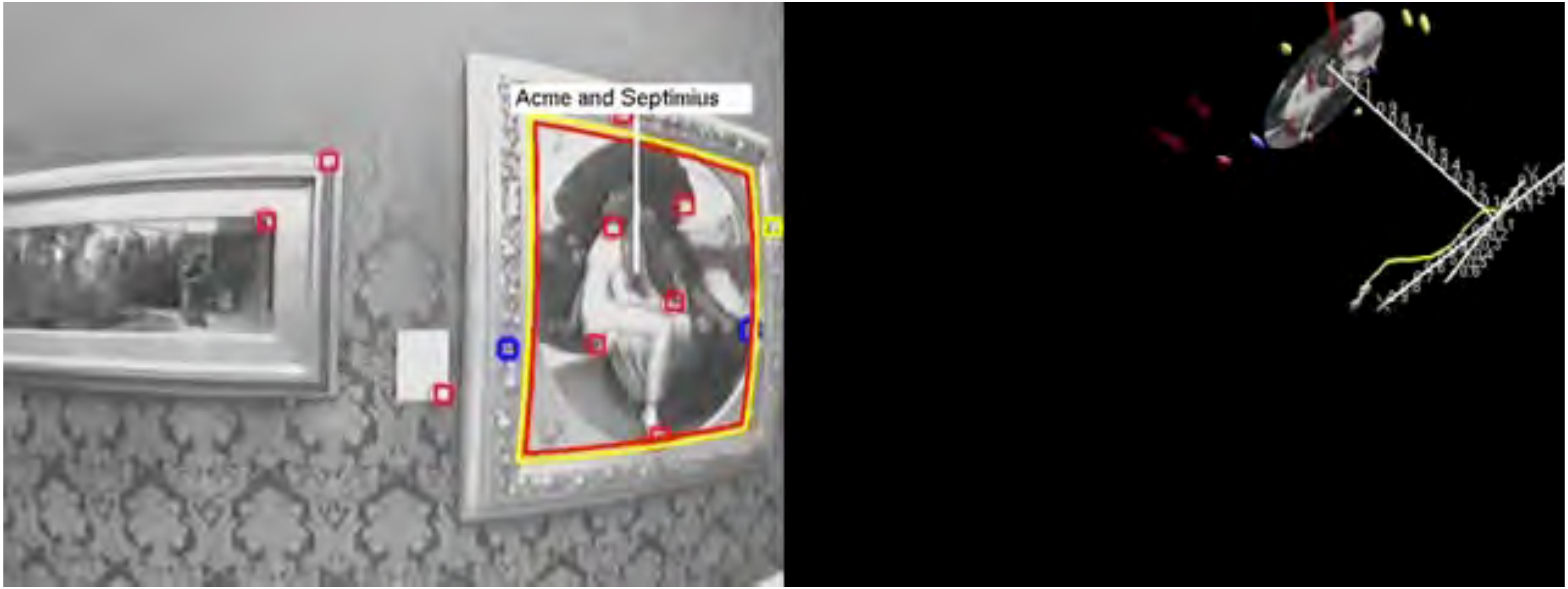
# EKF SLAM




Object tracking with MonoSLAM


- Sebastian Thrun, Wolfram Burgard, Dieter Fox. Probabilistic robotics.
- Davison, Andrew J., et al. "MonoSLAM: Real-time single camera SLAM." *IEEE Transactions on Pattern Analysis & Machine Intelligence* 6 (2007): 1052-1067.

# EKF SLAM

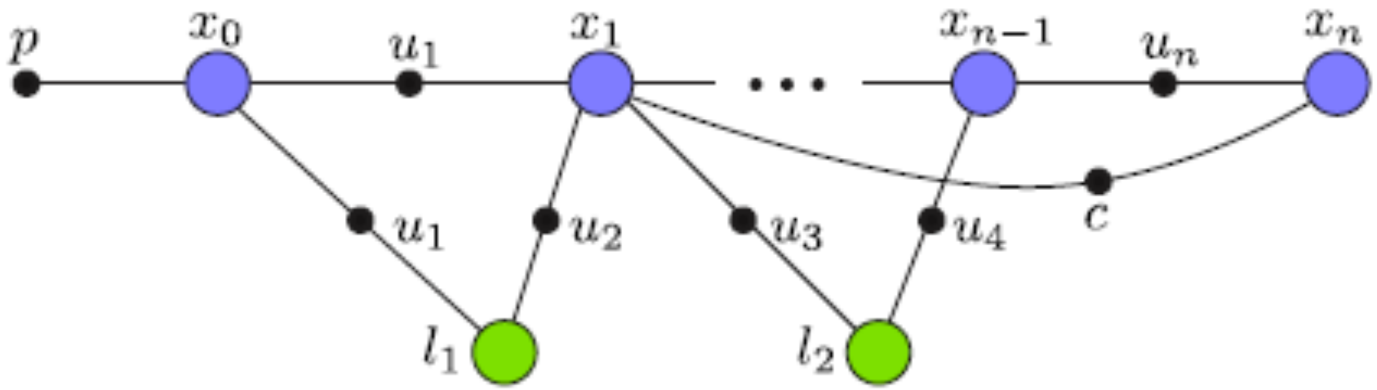



Object tracking with MonoSLAM


- Sebastian Thrun, Wolfram Burgard, Dieter Fox. Probabilistic robotics.
- Davison, Andrew J., et al. "MonoSLAM: Real-time single camera SLAM." *IEEE Transactions on Pattern Analysis & Machine Intelligence* 6 (2007): 1052-1067.

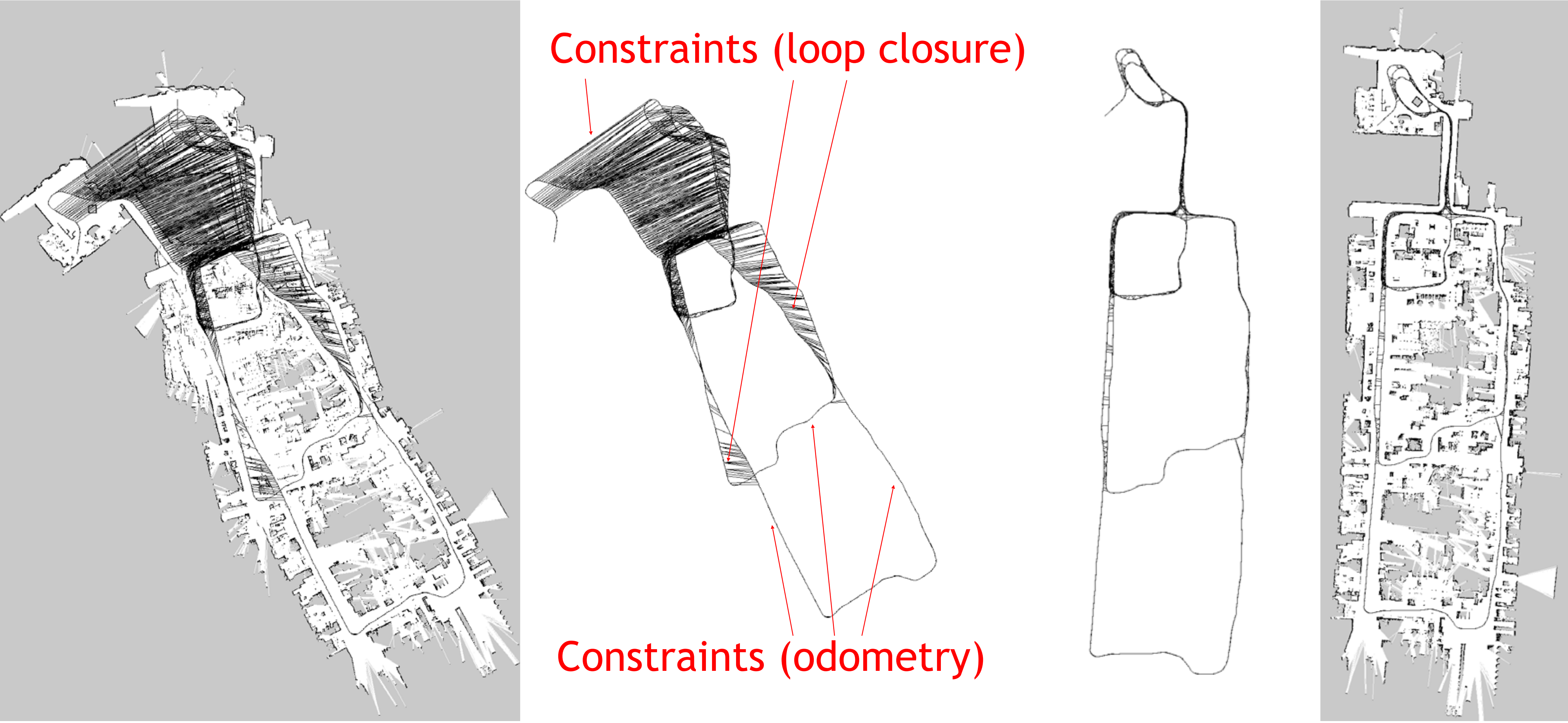
# Graph SLAM

- Use a **graph** to represent the problem
- Every node in the graph corresponds to a pose of the robot during mapping
- Every edge between two nodes corresponds to a spatial constraints (transformations from odometry/loop) between them




**Graph SLAM**: Build the graph and find a node configuration that minimize the the errors introduced by the constraints

# Graph SLAM Example




Before Graph SLAM

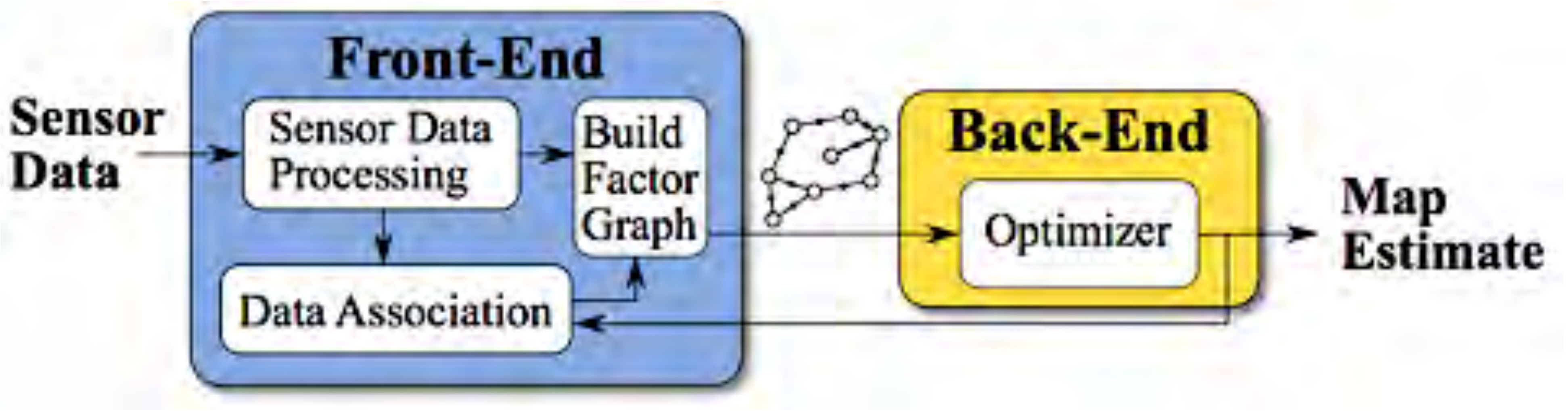
After Graph SLAM

# Graph SLAM

front-end: graph construction

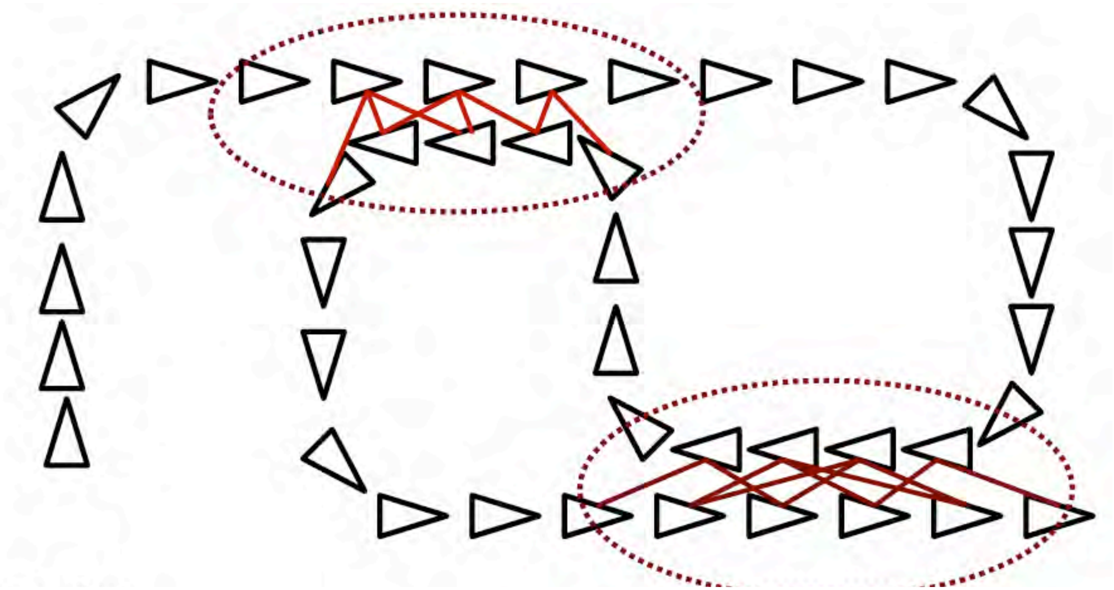
+

back-end: graph optimization



# Graph SLAM: Front-End

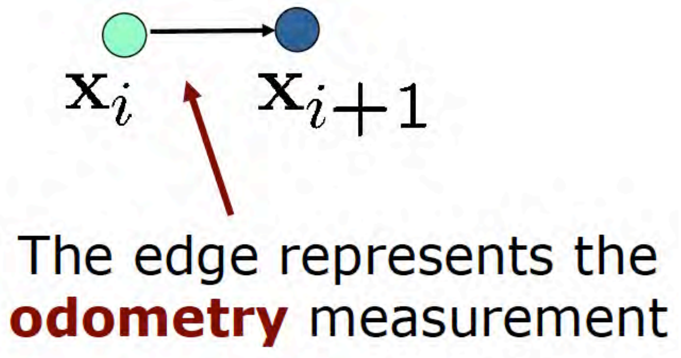
- Generate a graph with nodes & edges from sensor data
  - Each node is a 2D or 3D pose
  - A constraint/edge exists between 2 nodes
- 2 kinds of spatial constraints (transformations) as edges
  - **odometry constraints**
  - **observation constraints**



# Graph SLAM: Front-End

When the robot moves from $\mathbf{x}_i$ to $\mathbf{x}_{i+1}$

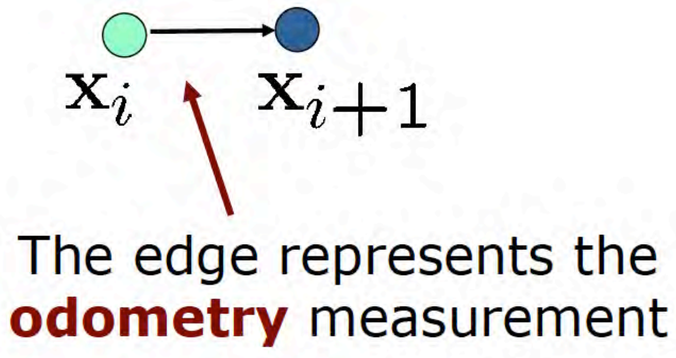
- **odometry constraints** between successive po
  - wheel/visual odometry
  - Iterative closest point (ICP)/scan matching for las



# Graph SLAM: Front-End

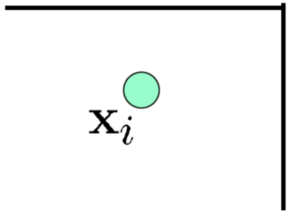
When the robot moves from $\mathbf{x}_i$ to $\mathbf{x}_{i+1}$

- **odometry constraints** between successive po
  - wheel/visual odometry
  - Iterative closest point (ICP)/scan matching for las

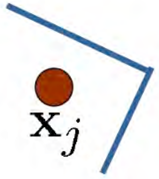



When the robot observes the same place from $\mathbf{x}_i$ and $\mathbf{x}_j$

- **observation constraints**
  - Observe a same landmark
  - when re-visiting a previously explored place




Measurement from $\mathbf{x}_i$

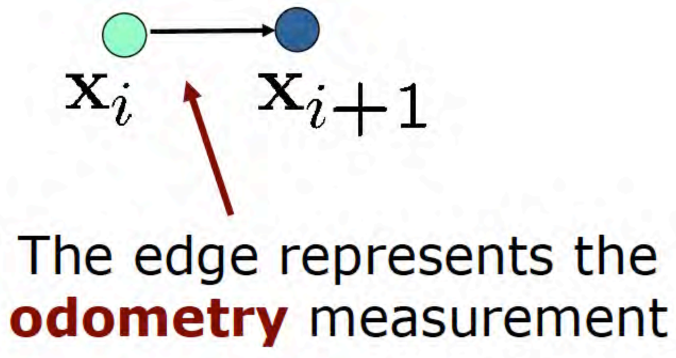



Measurement from $\mathbf{x}_j$

# Graph SLAM: Front-End

When the robot moves from $\mathbf{x}_i$ to $\mathbf{x}_{i+1}$

- **odometry constraints** between successive po
    - wheel/visual odometry
    - Iterative closest point (ICP)/scan matching for las




When the robot observes the same place fr$\mathbf{x}_i$n and $\mathbf{x}_j$om

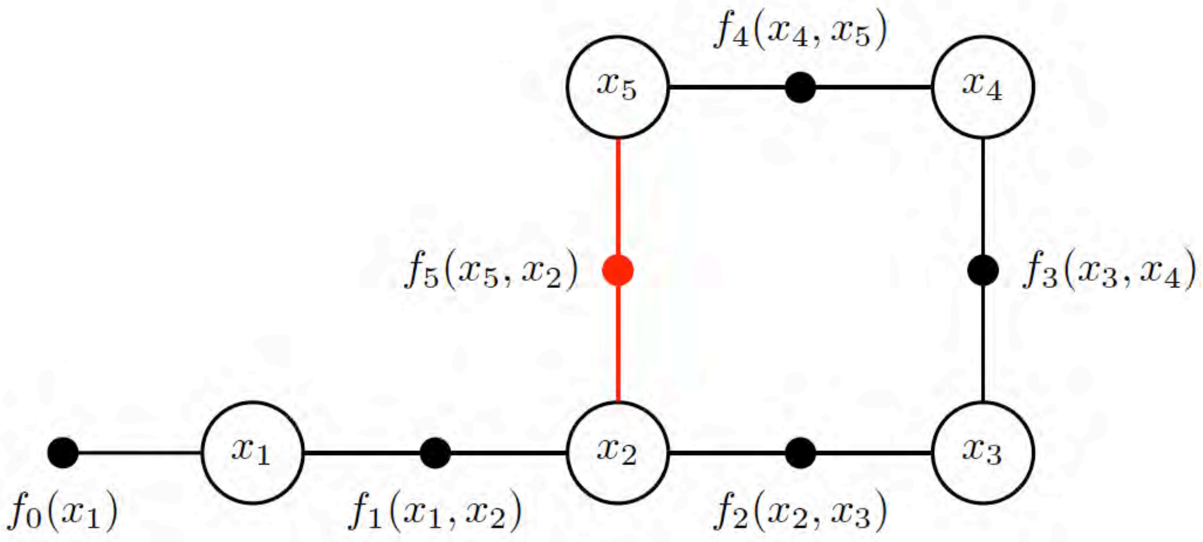
- **observation constraints**
    - Observe a same landmark
    - when re-visiting a previously explored place

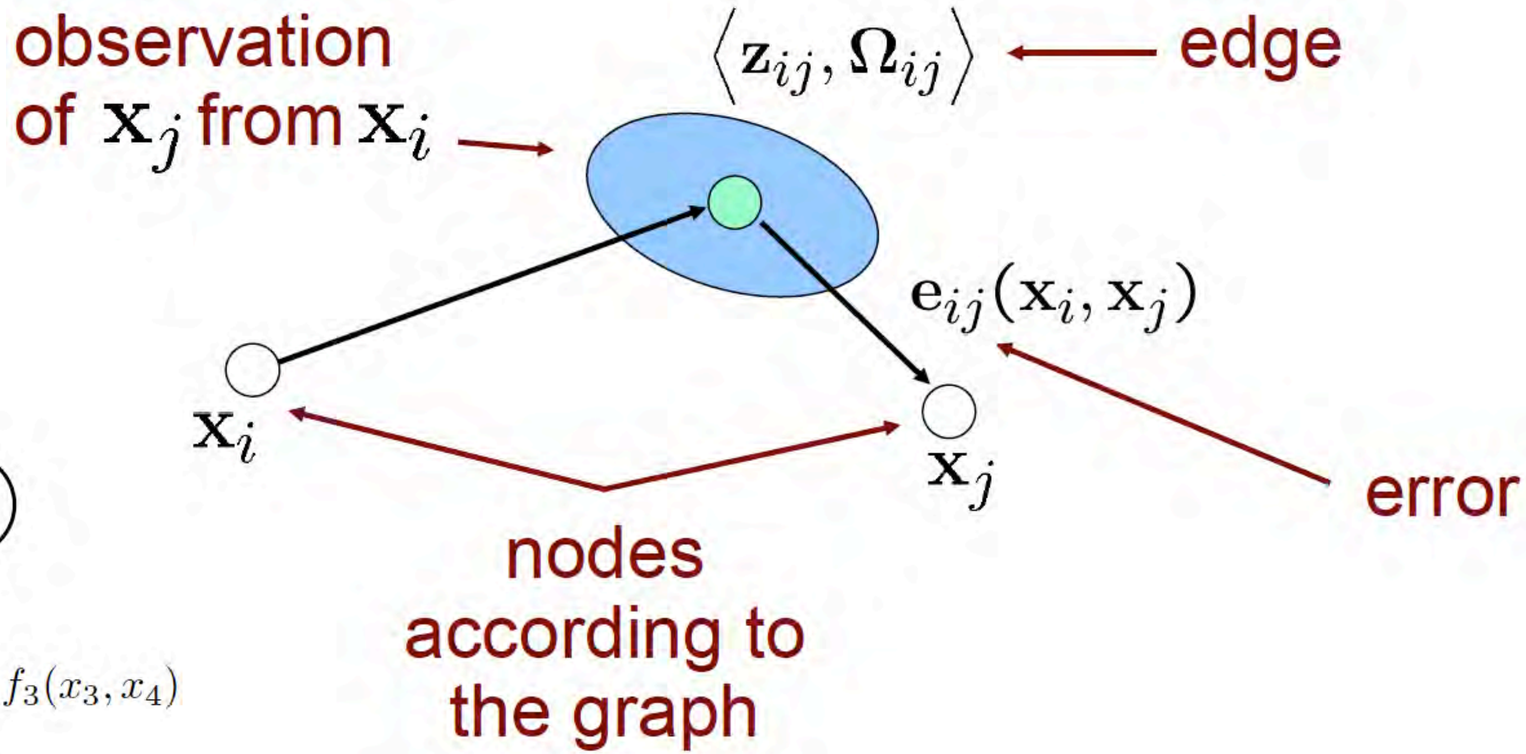

Measurement from $\mathbf{x}_i$

Measurement from $\mathbf{x}_j$

Edge represents the position of $\mathbf{x}_j$ seen from $\mathbf{x}_i$ based on the **observation**

# Graph SLAM: Front-End to Back-End



# Graph SLAM: Back-End

- Graph optimization: minimize the the errors introduced by the c
  - Error function for a single constraint

$$\mathbf{e}_{ij}(\mathbf{x}_i, \mathbf{x}_j) = \text{t2v}(\underbrace{\mathbf{Z}_{ij}^{-1}}_{\text{measurement}}\underbrace{(\mathbf{X}_i^{-1}\mathbf{X}_j)}_{\boldsymbol{x}_j \text{ referenced w.r.t. } \boldsymbol{x}_i})$$

  - Error takes a value of zero if

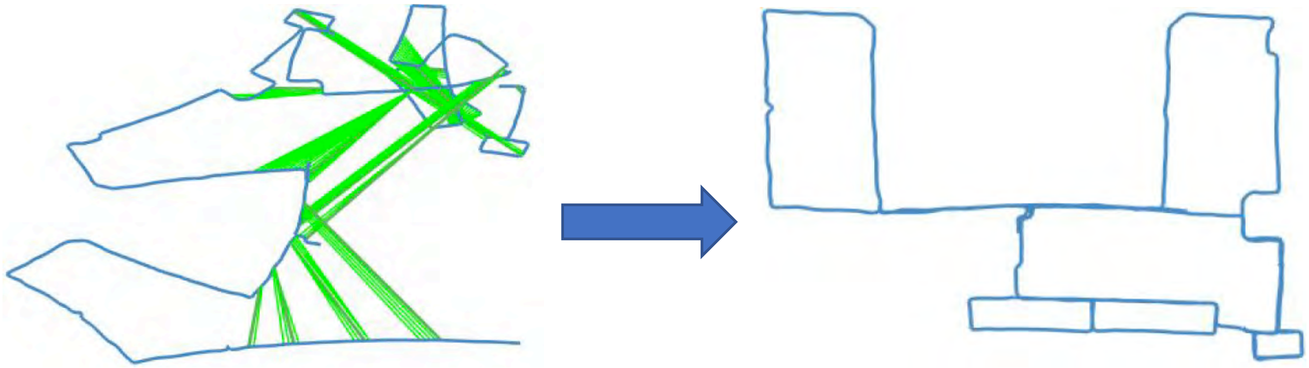
$$\mathbf{Z}_{ij} = (\mathbf{X}_i^{-1}\mathbf{X}_j)$$

  - **Goal:** $\mathbf{x}^* = \underset{\mathbf{x}}{\text{argmin}} \sum_{ij} \mathbf{e}_{ij}^T \boldsymbol{\Omega}_{ij} \mathbf{e}_{ij}$

- Solve the optimization: Gauss-Newton/Levenberg-Marquardt

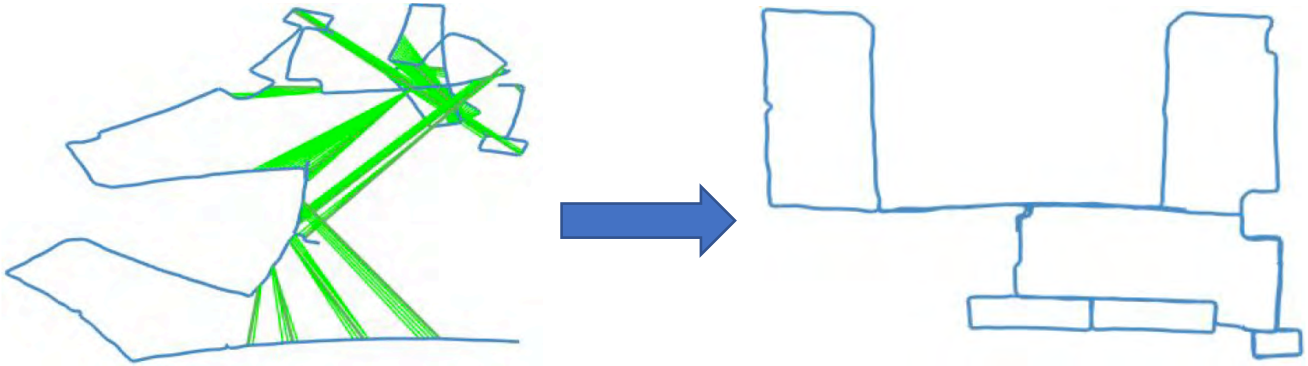
# Graph SLAM: Back-End

Back-end open-source tools:
g2o: https://github.com/RainerKuemmerle/g2o
iSAM: http://people.csail.mit.edu/kaess/isam/



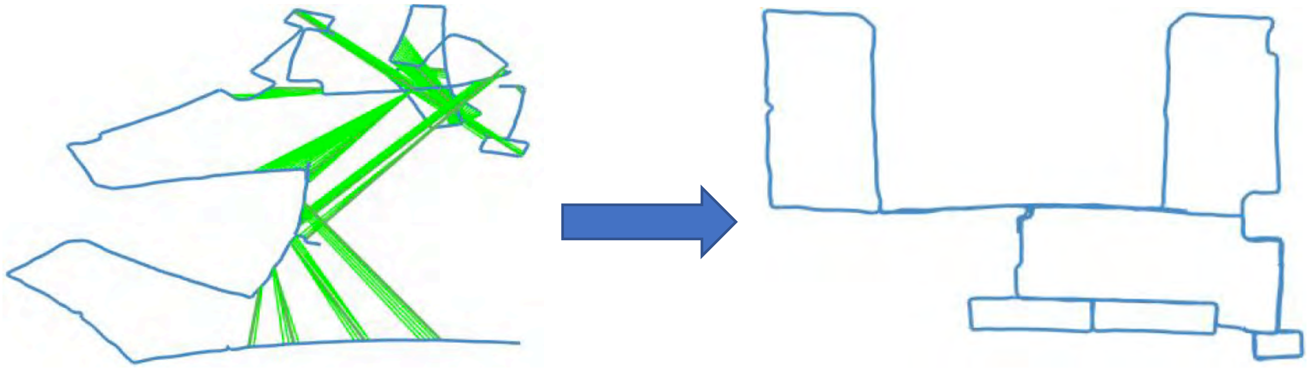
# Graph SLAM: Back-End

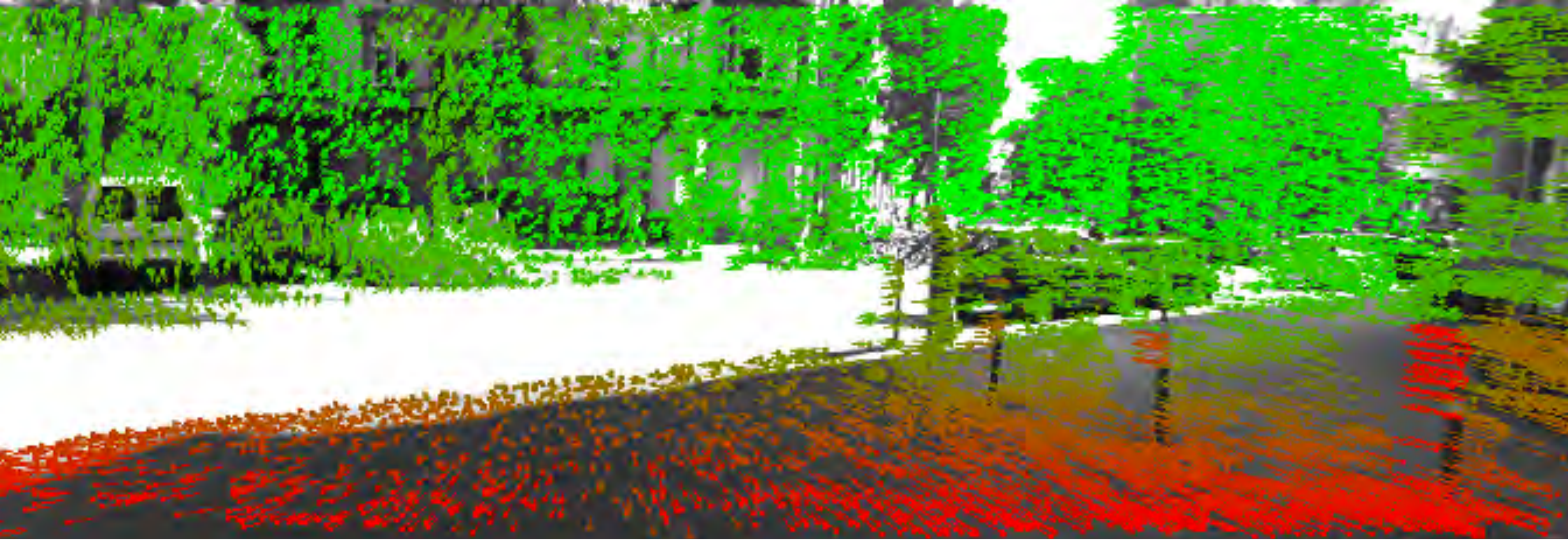
Back-end open-source tools:
g2o: https://github.com/RainerKuemmerle/g2o
iSAM: http://people.csail.mit.edu/kaess/isam/



# Graph SLAM: Back-End

Back-end open-source tools:
g2o: https://github.com/RainerKuemmerle/g2o
iSAM: http://people.csail.mit.edu/kaess/isam/





# Visual Odometry & SLAM

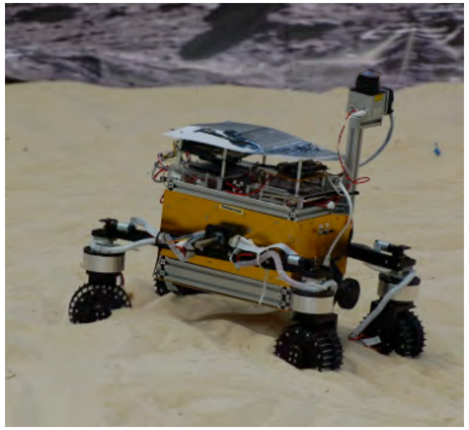
# Visual Odometry/SLAM: Applications

# Visual Odometry/SLAM: Applications

# Visual Odometry/SLAM: Applications

# Why Vision based Positioning/VO/SLAM

- camera: light-weight, low-cost, rich information
- flexible: mobile, flying, underwater robots
- pervasive: phones, cars, laptops
- more accurate than wheel odometry
- no wheel slip

# Types – VO/Visual SLAM

- Geometry based Methods: extract geometric constraints from imagery to estimate motion
  - **Sparse feature based methods**
  - Direct/Dense methods

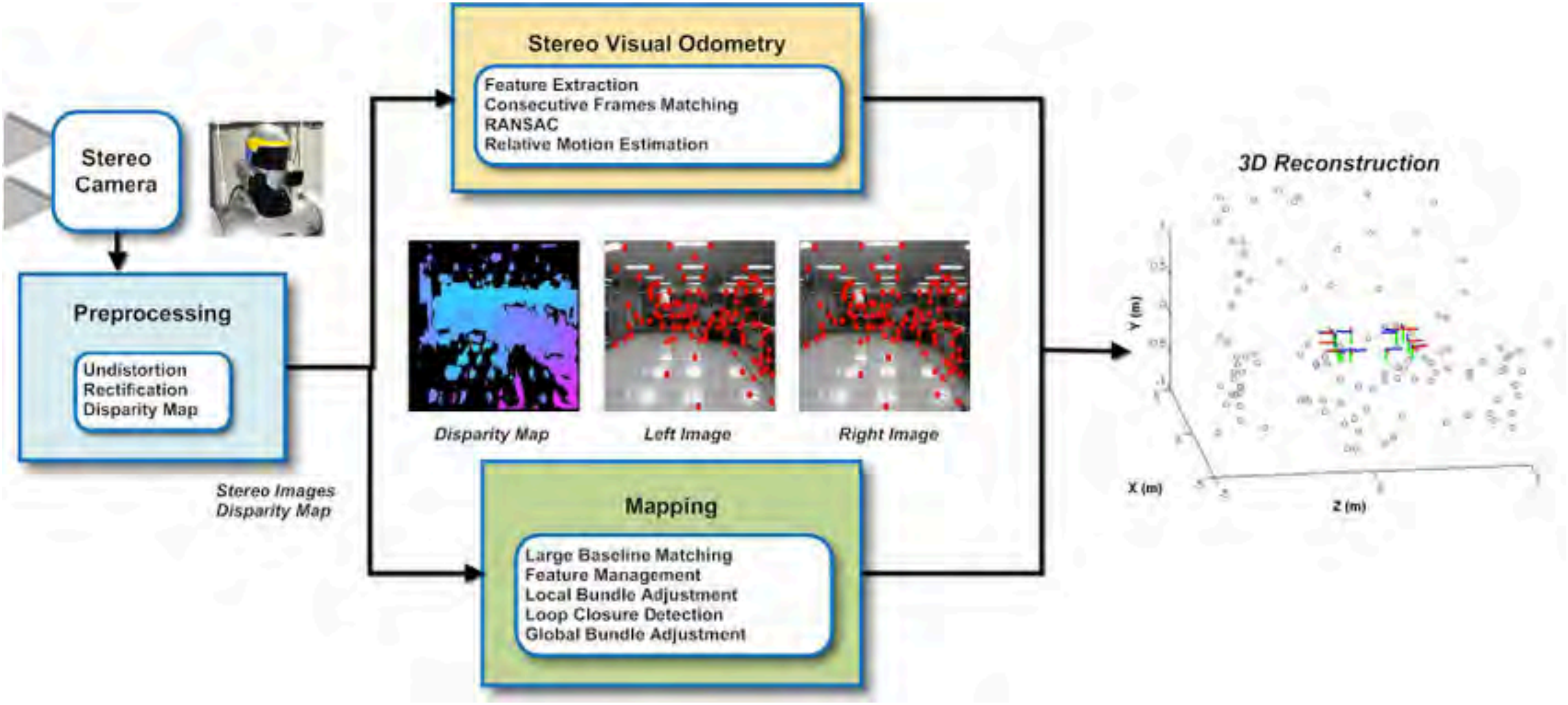
- Learning based Methods: use machine learning techniques without explicitly applying geometric theory

# Types – VO/Visual SLAM

- Geometry based Methods: extract geometric constraints from imagery to estimate motion
  - **Sparse feature based methods**
  - Direct/Dense methods

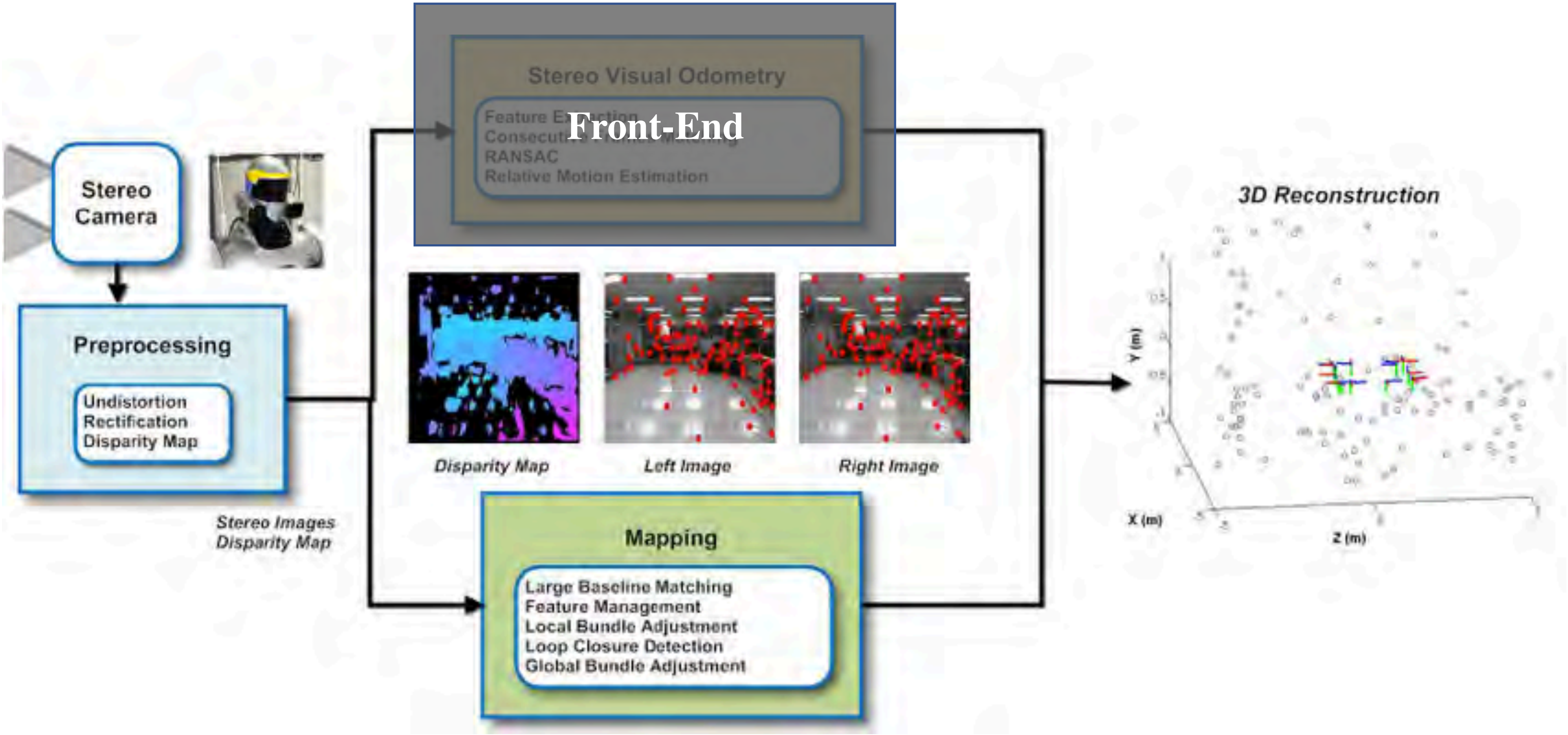
- Learning based Methods: use machine learning techniques without explicitly applying geometric theory

# Visual SLAM Pipeline



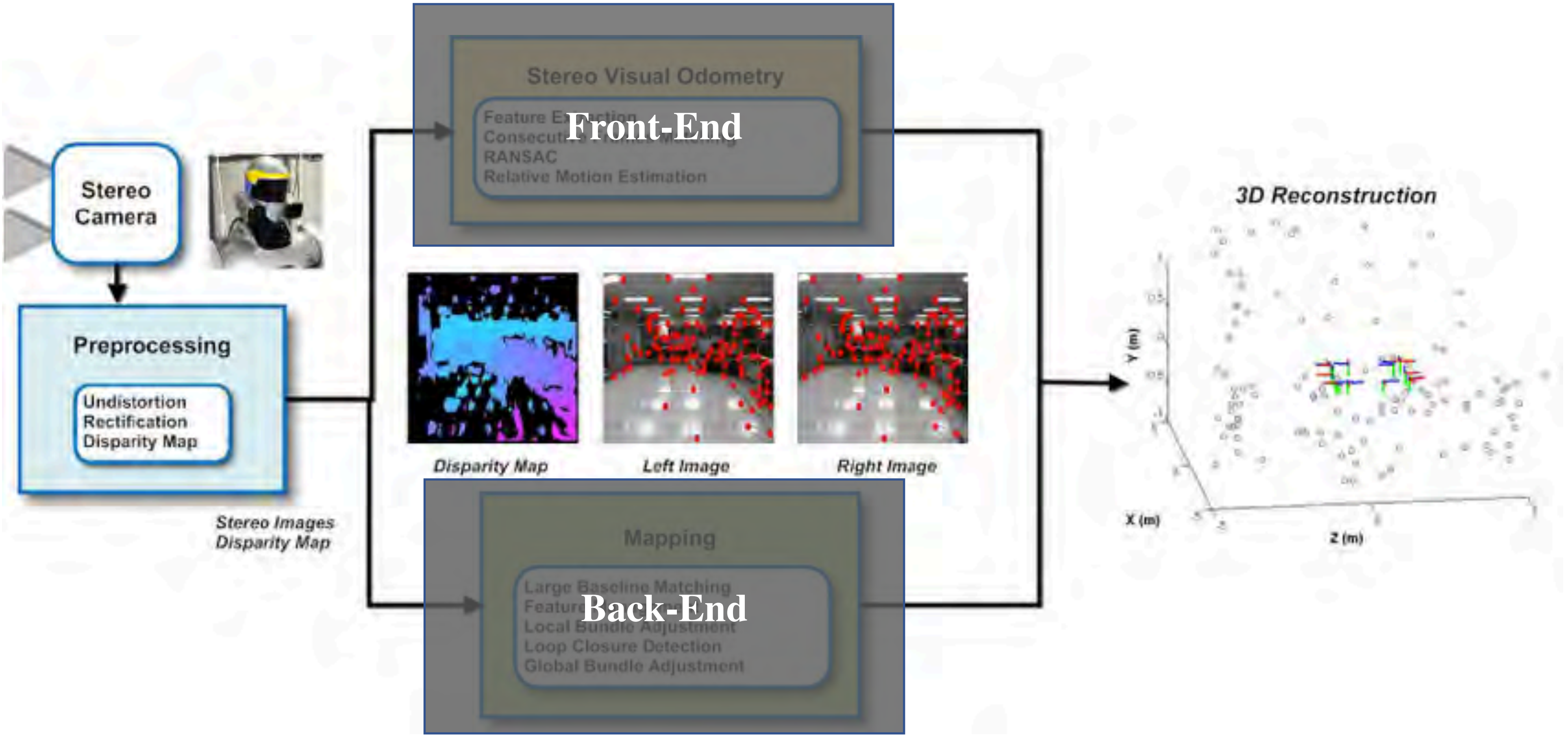

Alcantarilla et al. AR’13

# Visual SLAM Pipeline



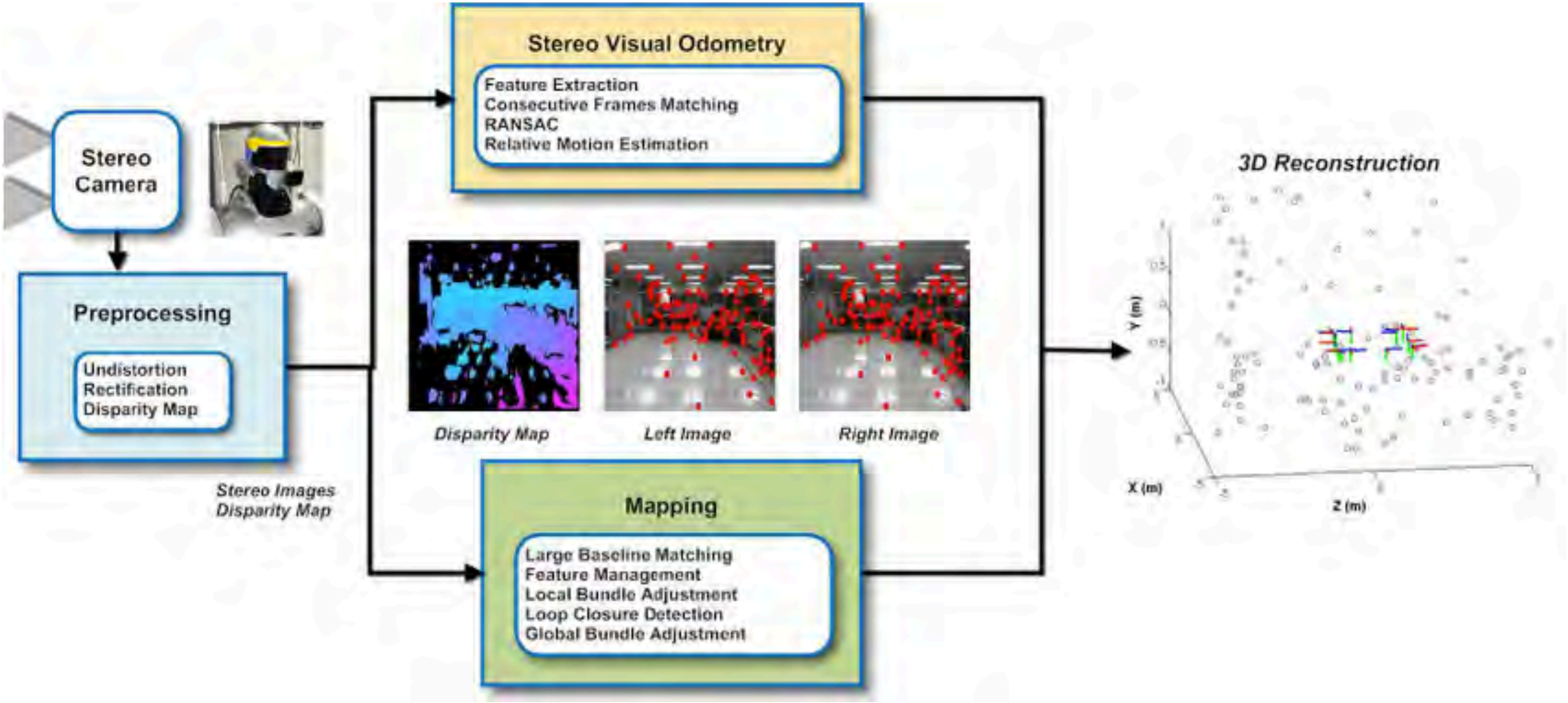

Alcantarilla et al. AR'13

# Visual SLAM Pipeline



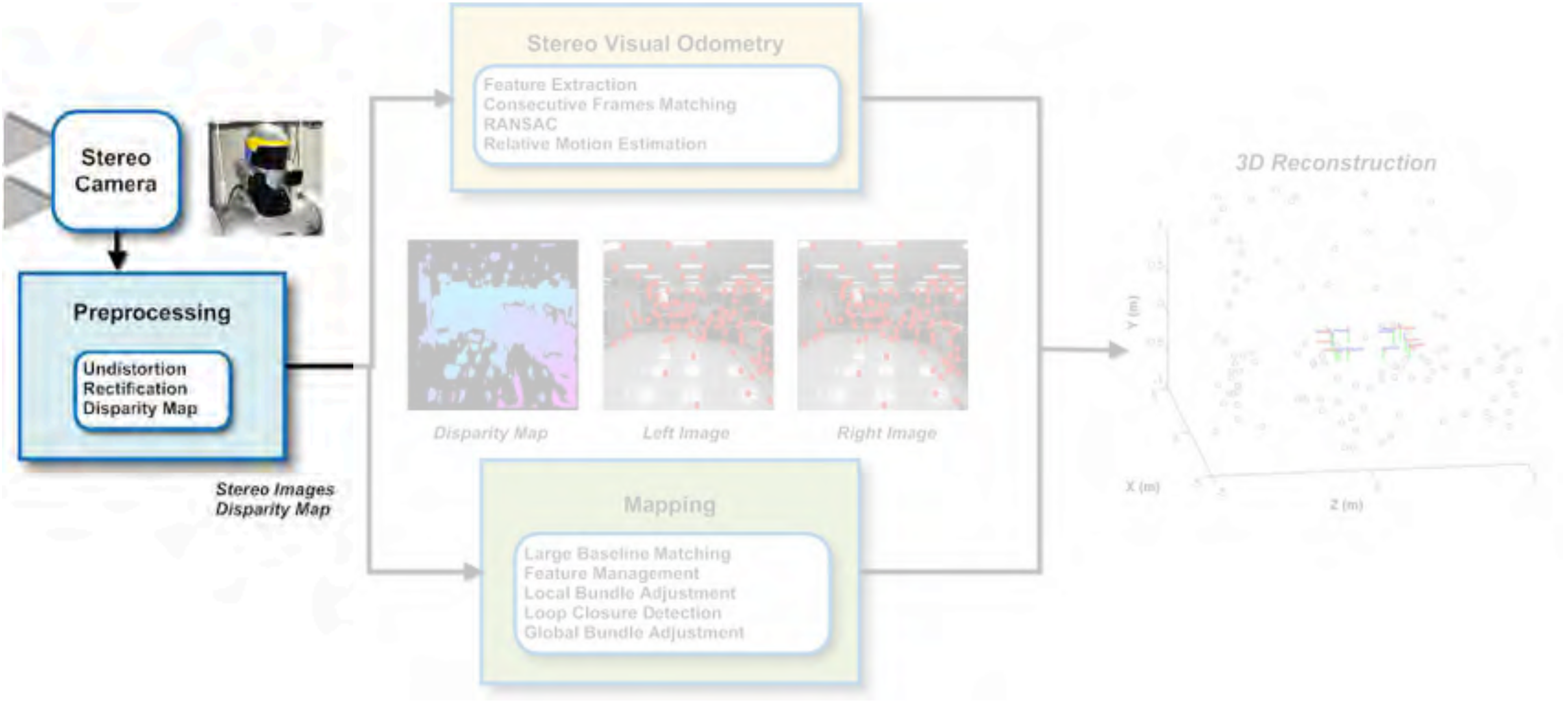

Alcantarilla et al. AR’13

# Visual SLAM Pipeline




Alcantarilla et al. AR'13

# Visual SLAM Pipeline

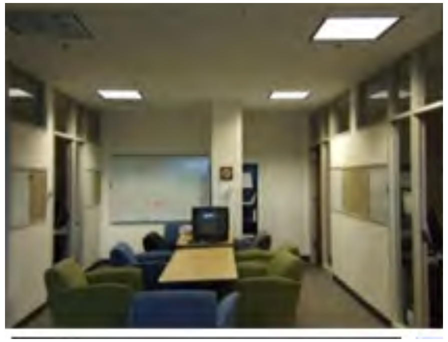



Alcantarilla et al. AR'13

# Camera Sensors

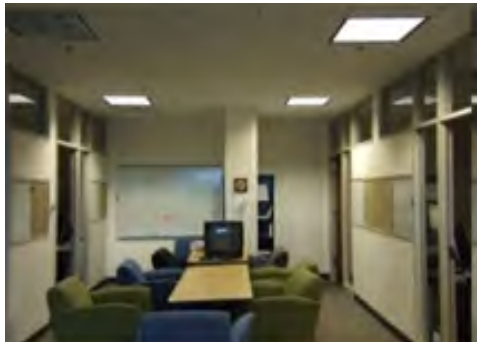
- >=30Hz images

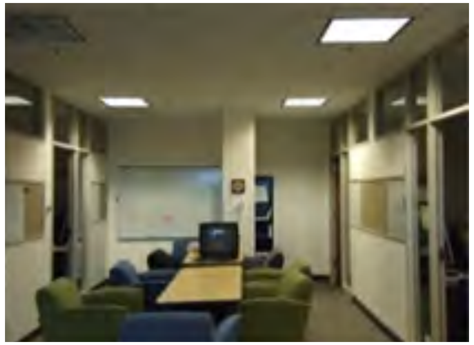
Monocular camera



Stereo Camera
- KinectXtion
- PRO LIVE

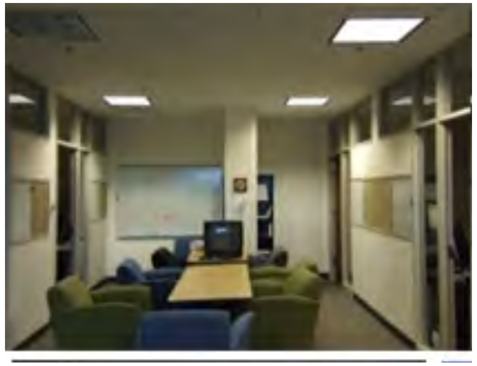
~0.5 m to ~ 20 m
(depends on baseline)

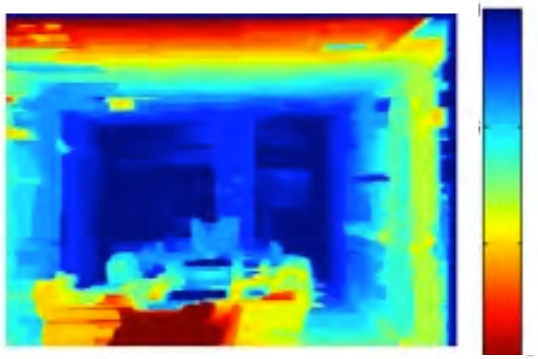




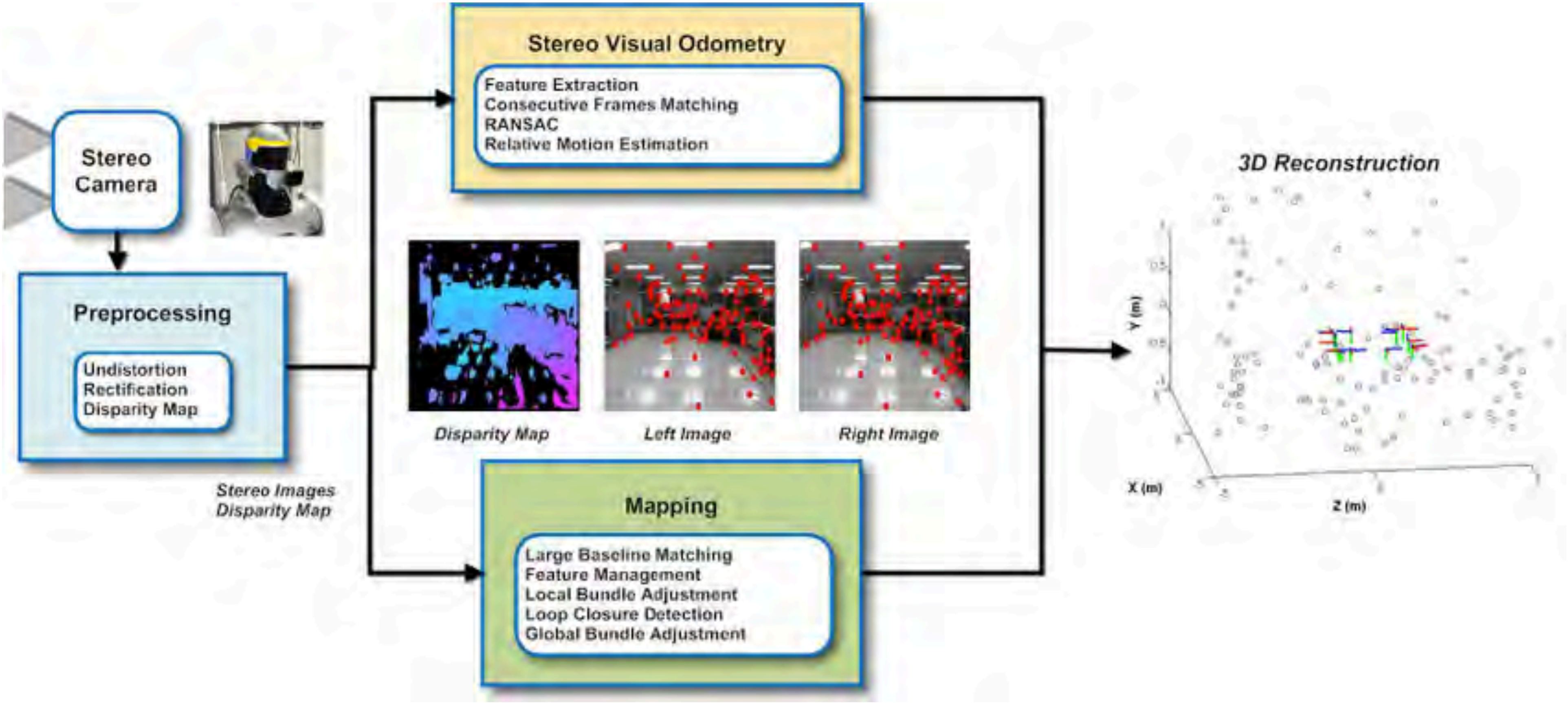
RGB-D Sensors
- Kinect
- Xtion PRO LIVE

Range: ~0.5 m to ~5 m



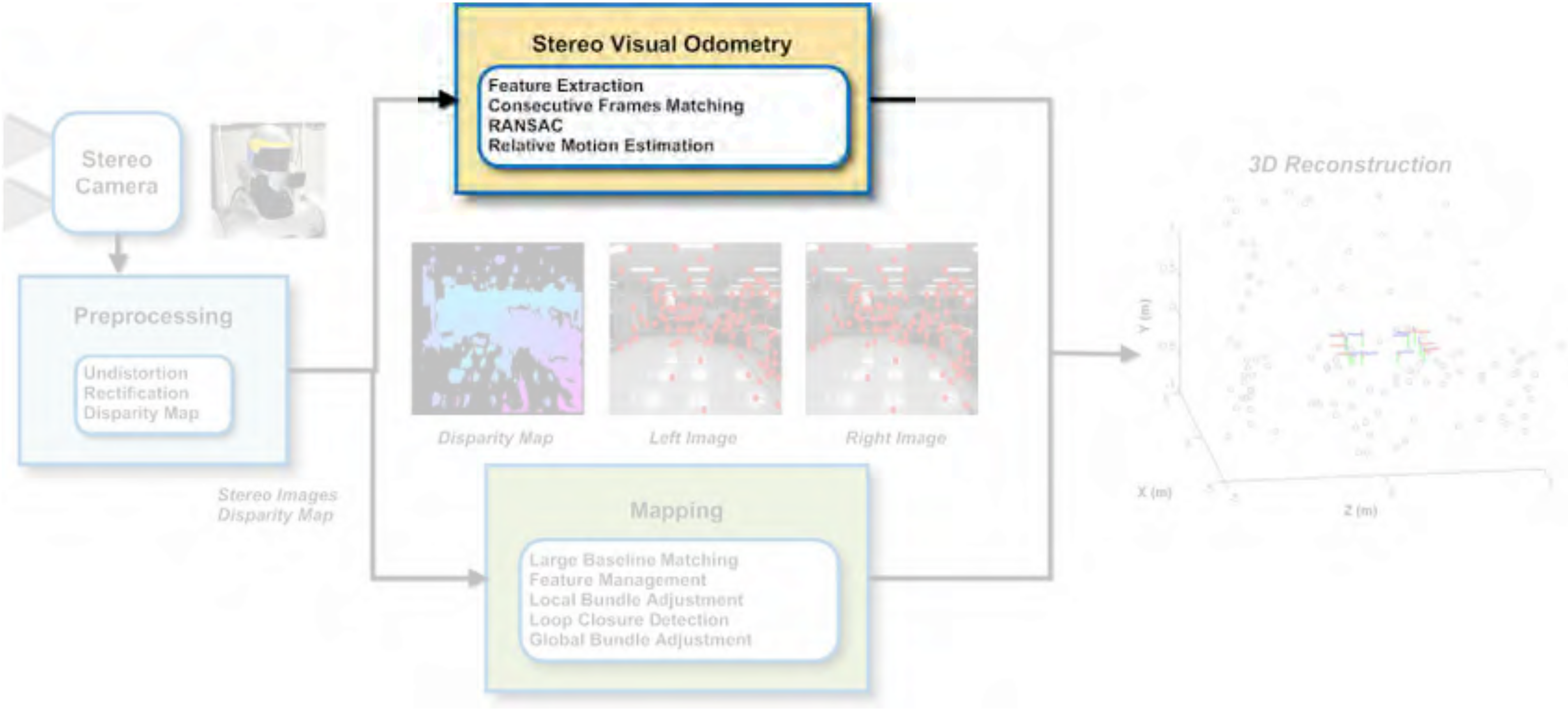


# Visual SLAM Pipeline




Alcantarilla et al. AR’13

# Visual SLAM Pipeline



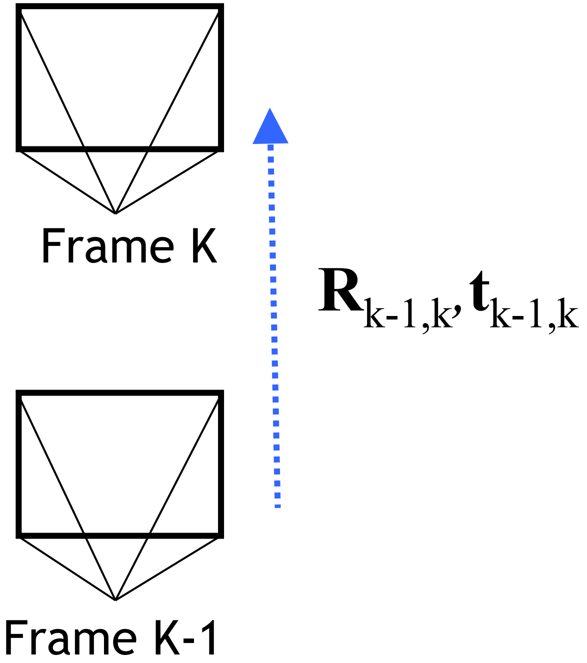

Alcantarilla et al. AR'13

# Visual Odometry

- motion: changes on the pixels of the images
- changes: estimate motion, pose (x, y, z, roll, pitch, yaw)

move to see, see to move

Images from KITTI Dataset



# Visual Odometry

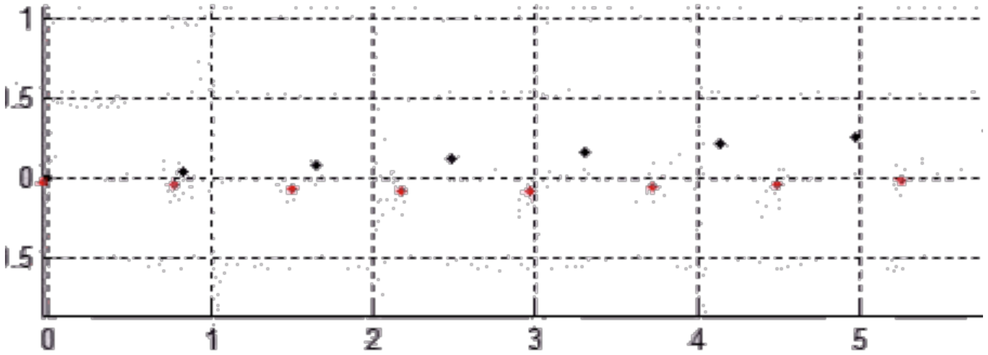
- motion: changes on the images
- changes: estimate motion, pose (x, y, z, roll, pitch, yaw)

frame-to-frame motion over time

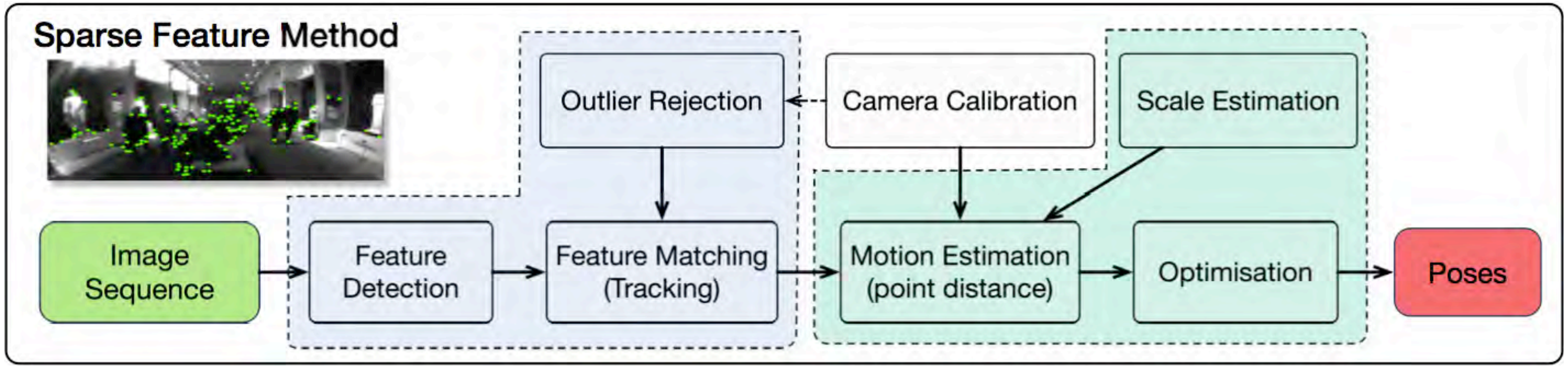
Input: Video (Image Sequence)

VO

Result: Visual Odometry



# Sparse Feature based VO

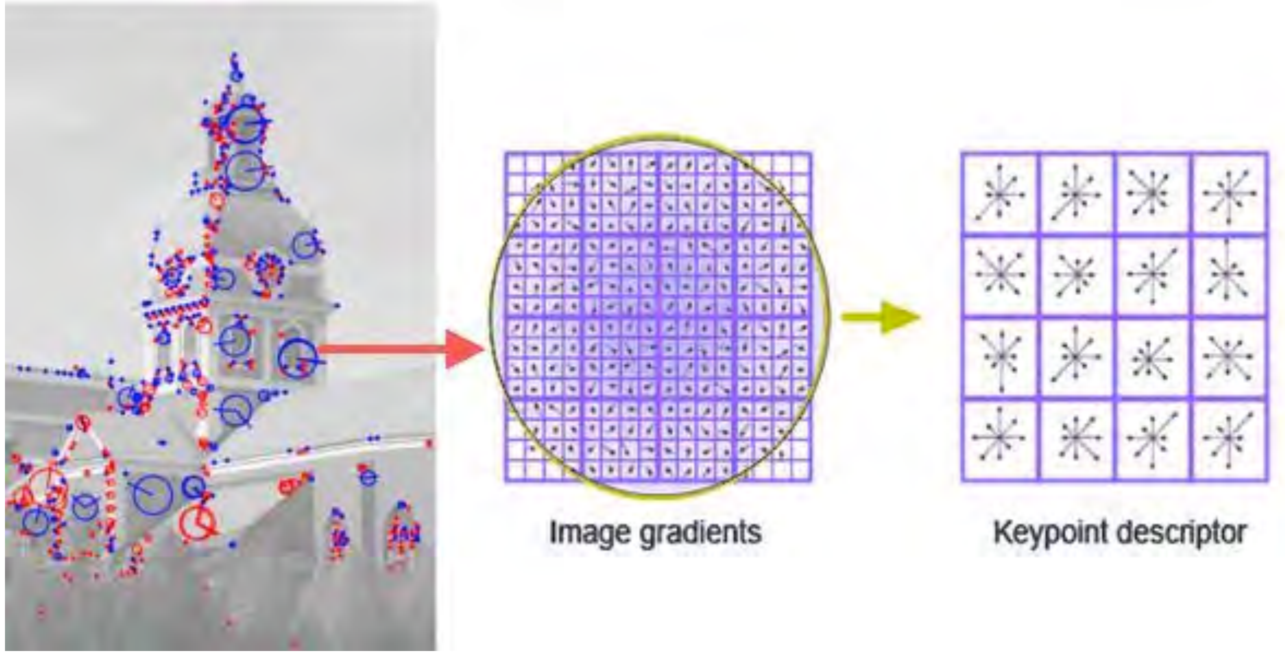


# Feature Extraction

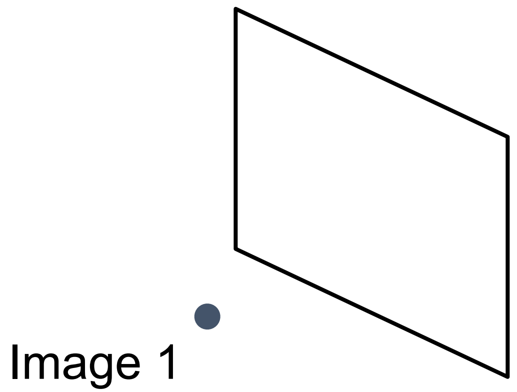
## 2D Features:

- FAST
- GFTT
- BRISK
- ORB
- BRIEF
- SIFT
- SURF

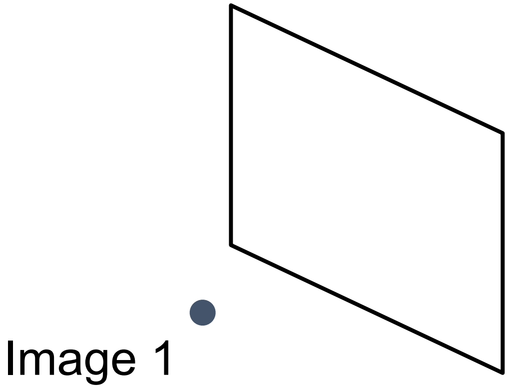
See OpenCV for more…



# Sparse Feature based Methods – F2F



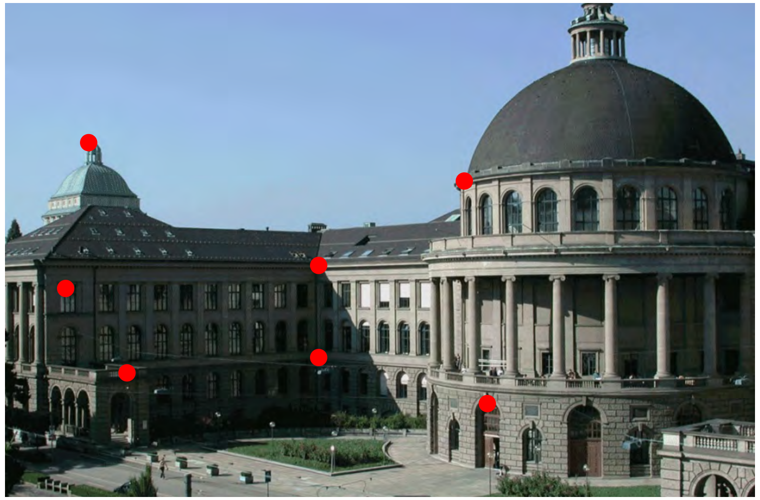
# Sparse Feature based Methods – F2F



# Sparse Feature based Methods – F2F

slide courtesy of Davide Scaramuzza

# Sparse Feature based Methods – F2F



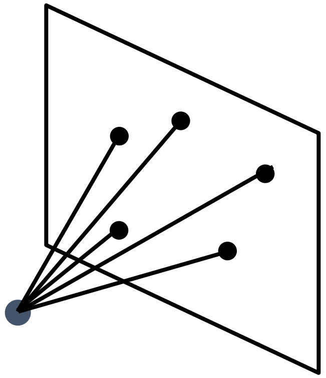


Image 1

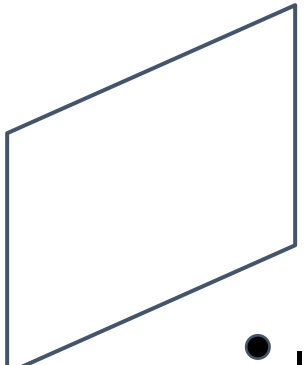
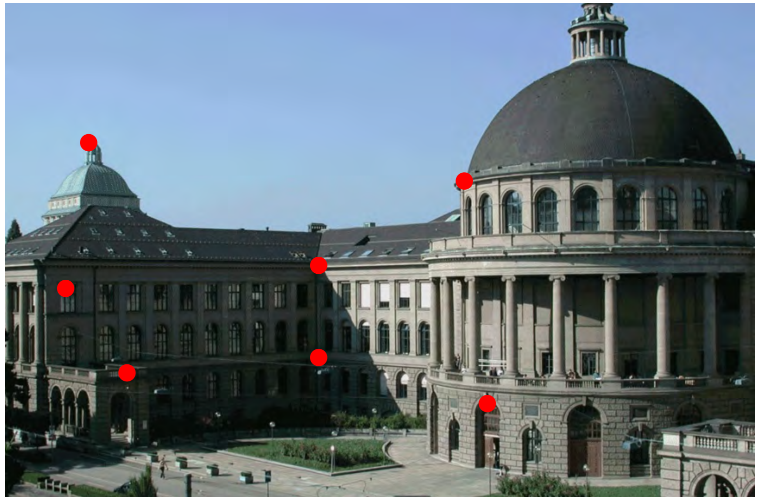
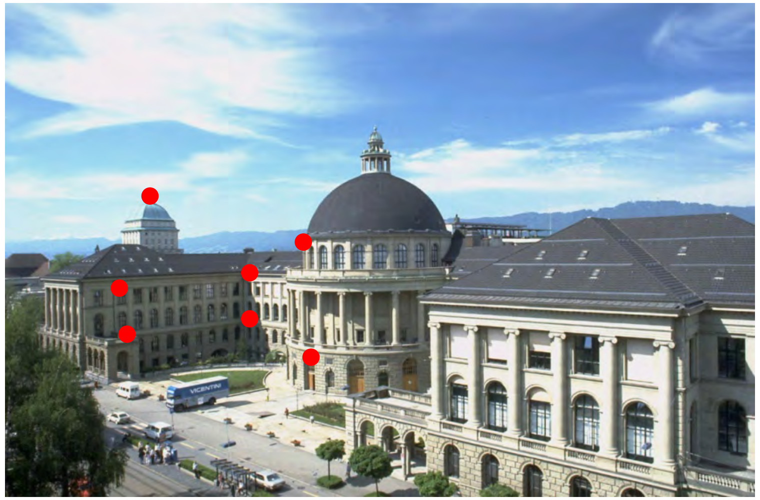
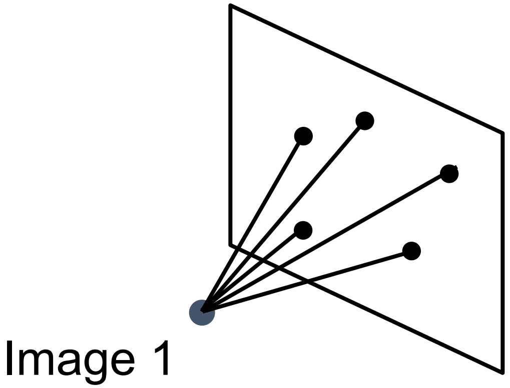


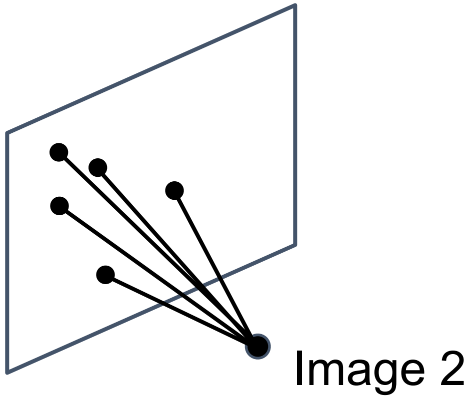
Image 2

# Sparse Feature based Methods – F2F



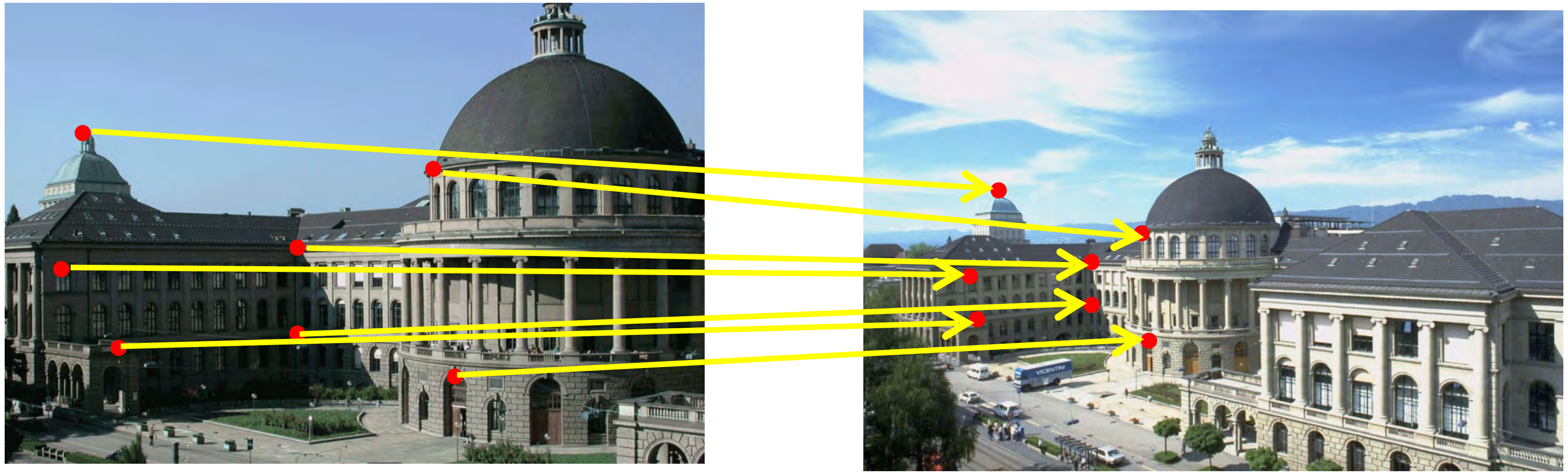




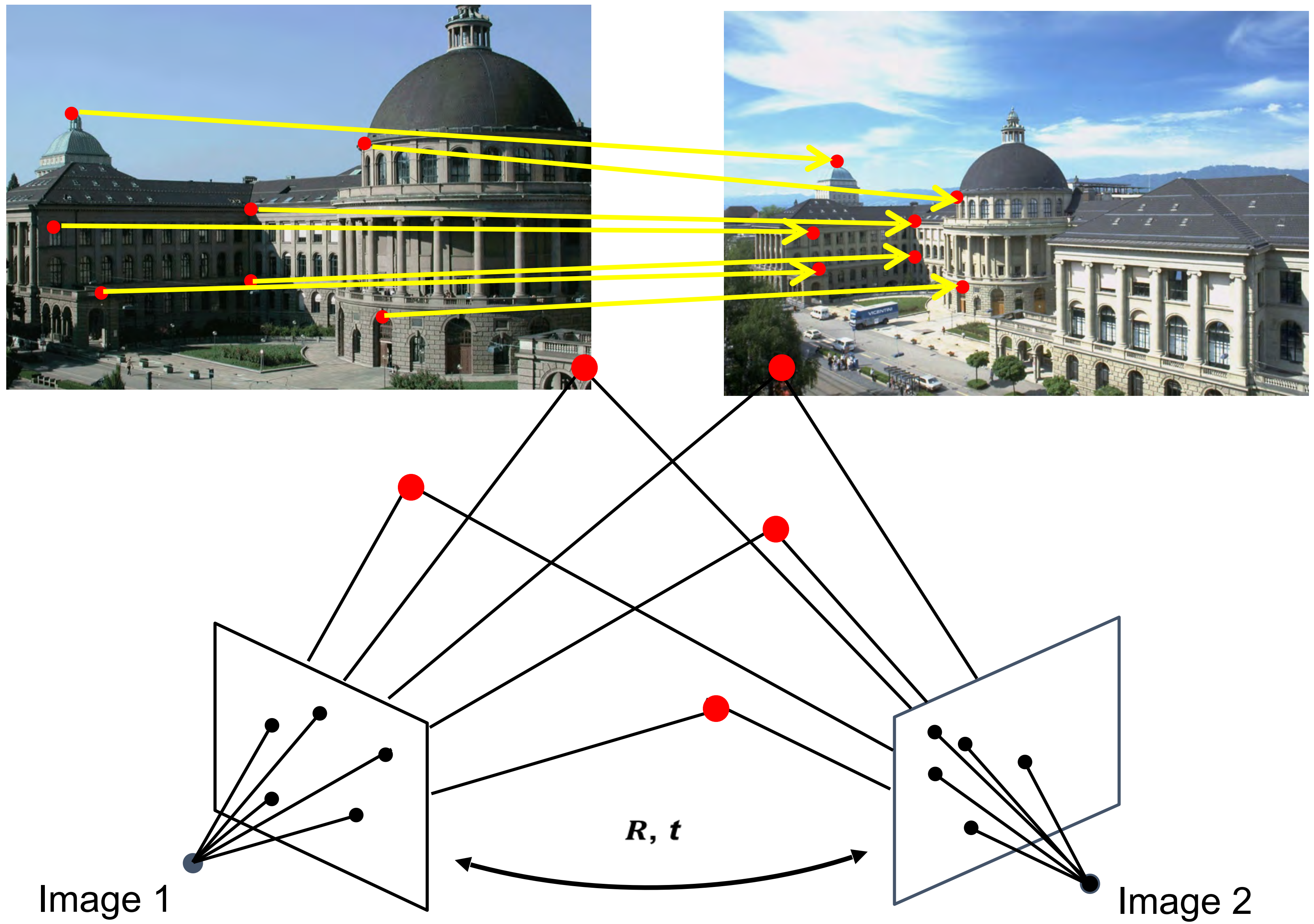
slide courtesy of Davide Scaramuzza

# Sparse Feature based Methods – F2F



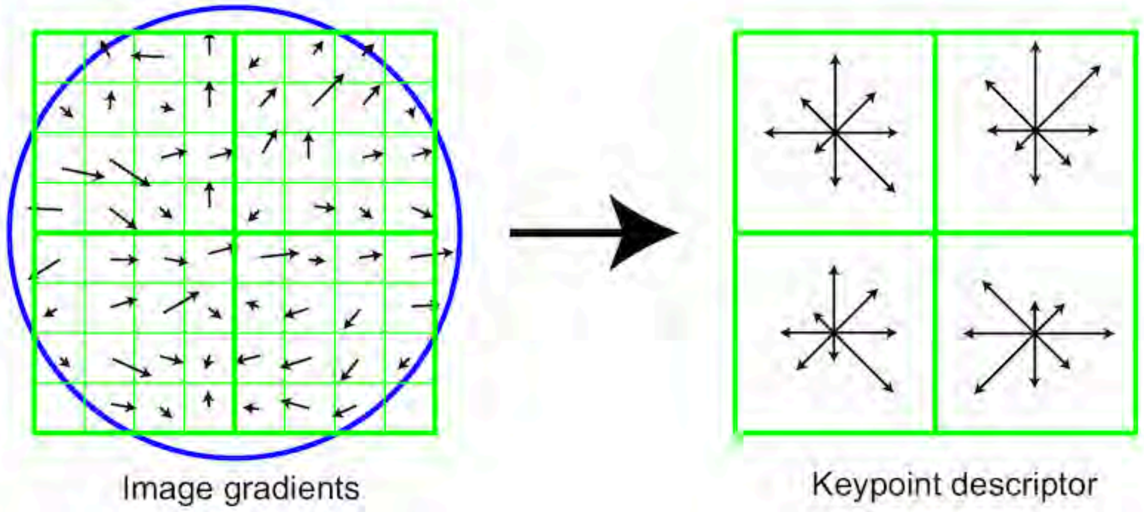
slide courtesy of Davide Scaramuzza

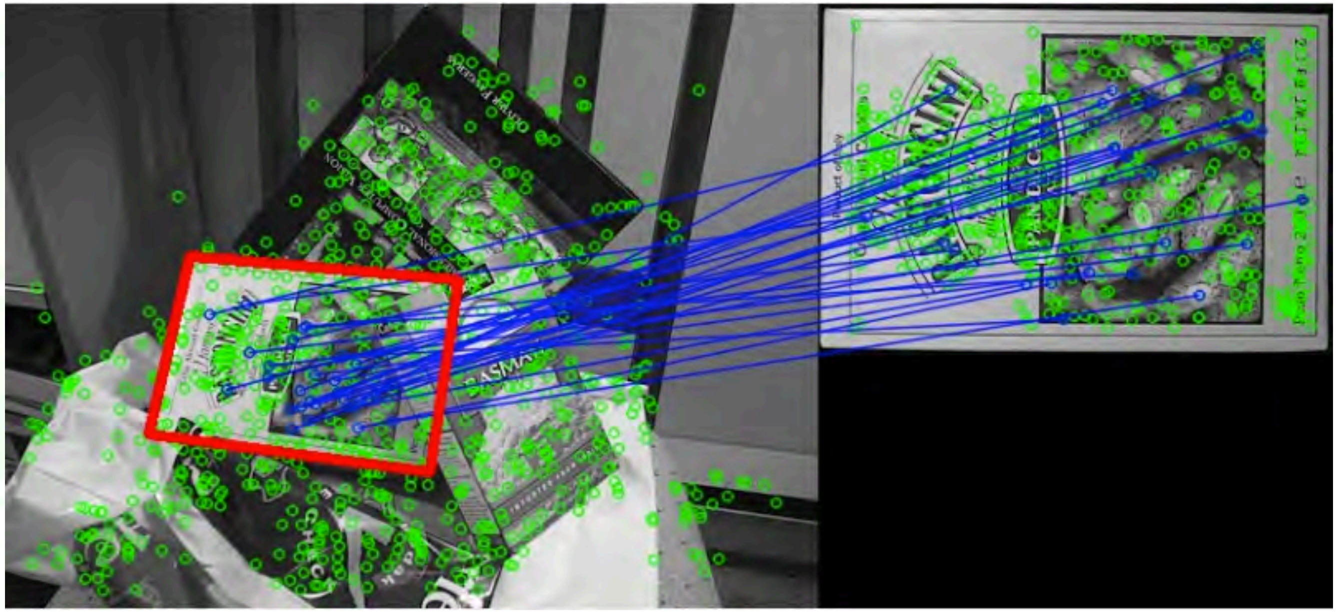
# Sparse Feature based Methods – F2F



# Feature Matching

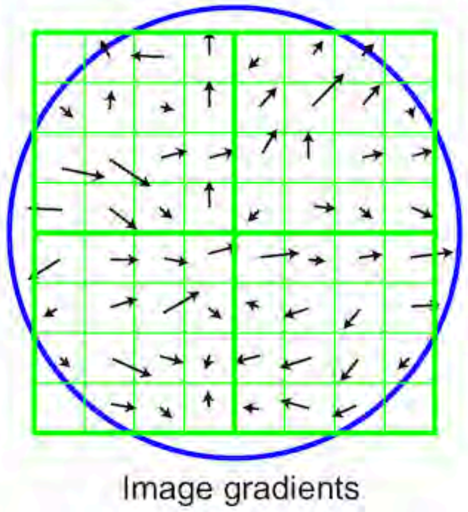
- Correspondences between frames using feature descriptors

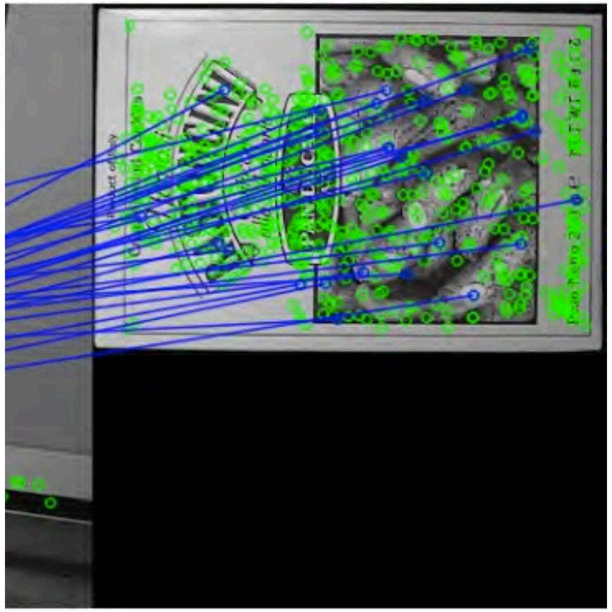


# Feature Matching

- Correspondences between frames using feature descriptors



# Sparse Feature based Methods

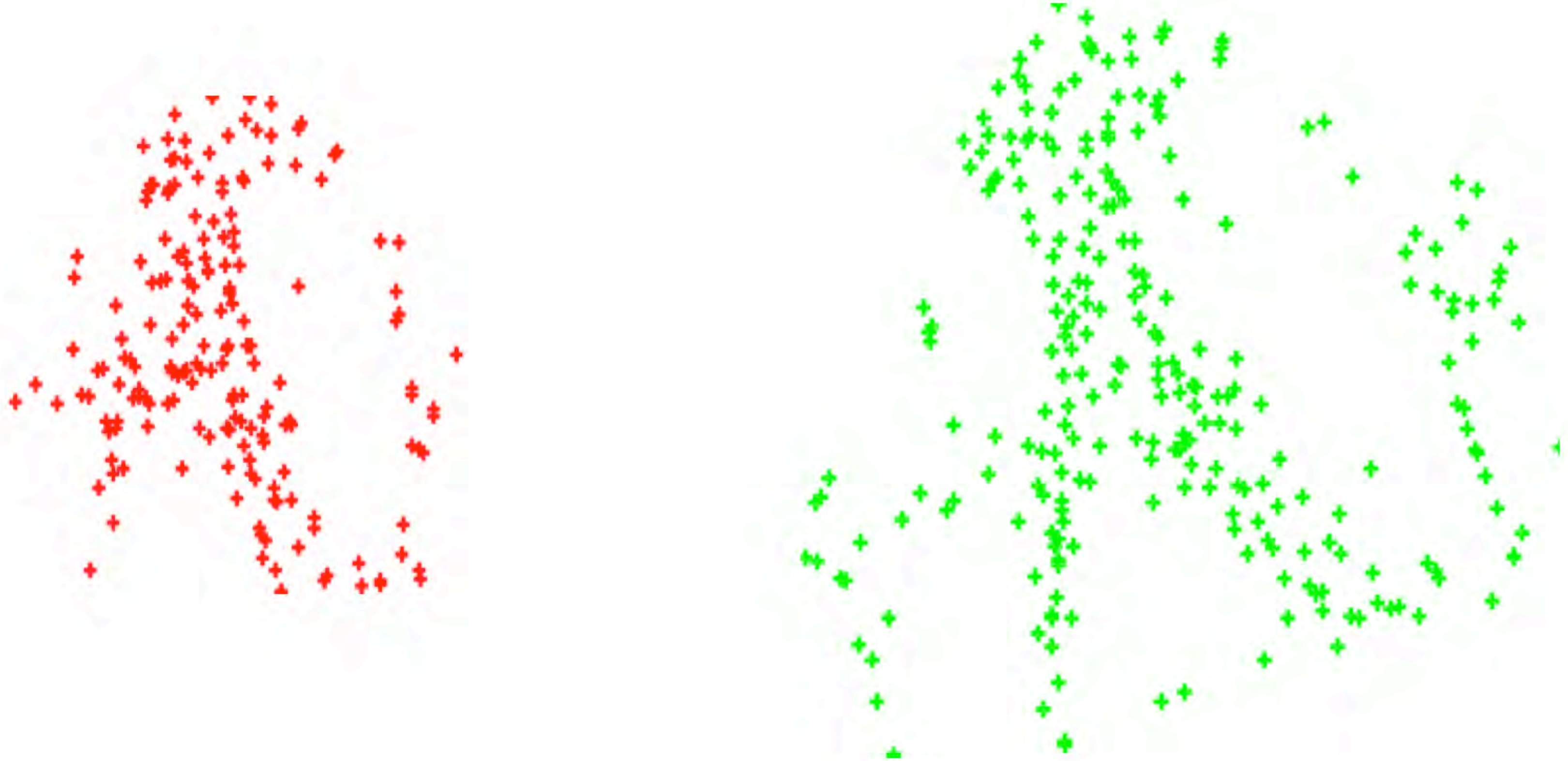
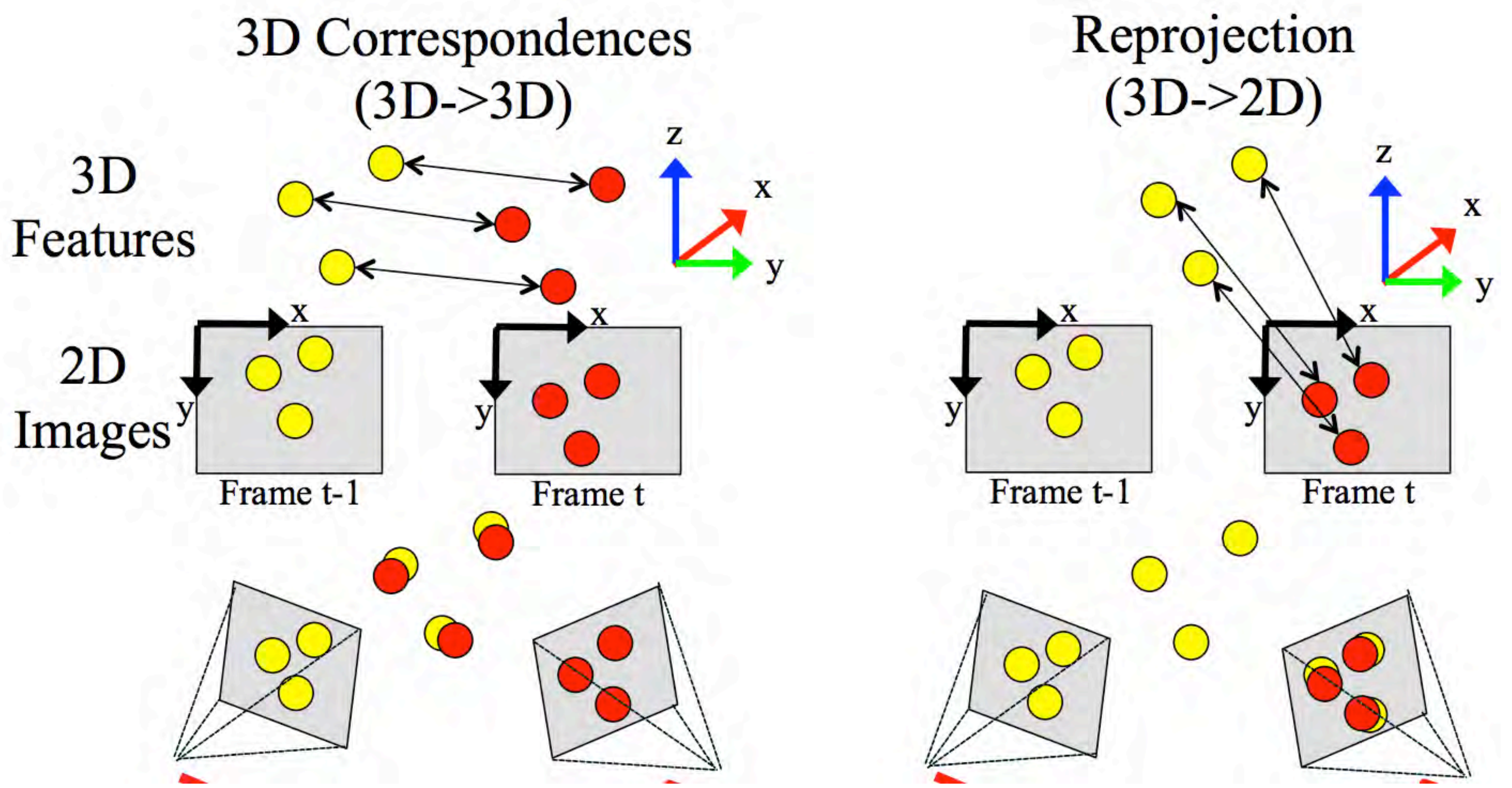


Image is reduced to a sparse set of **keypoints**
Usually matched with feature **descriptors**

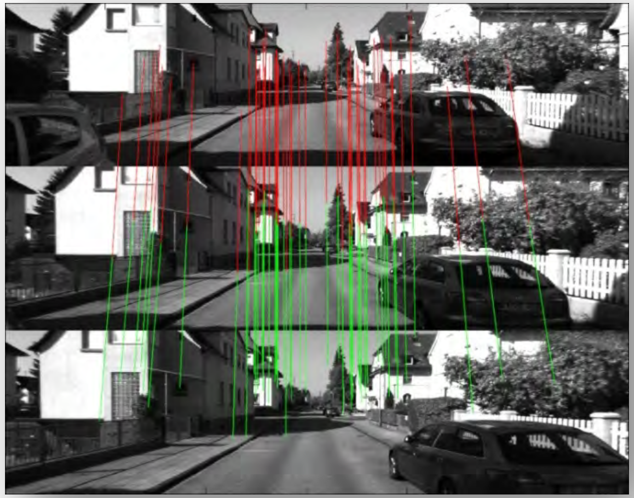
# Transformation Estimation: Essential Matrix

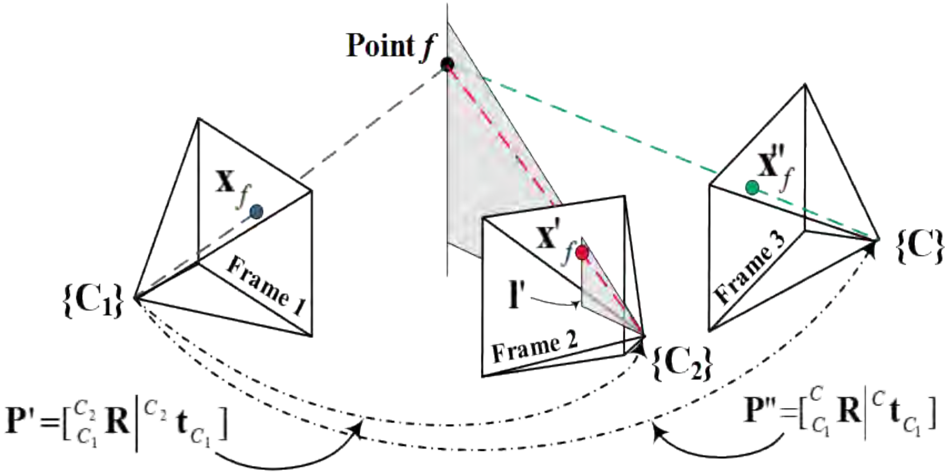



slide courtesy of Labbé

# Motion

compute camera trajectory incrementally

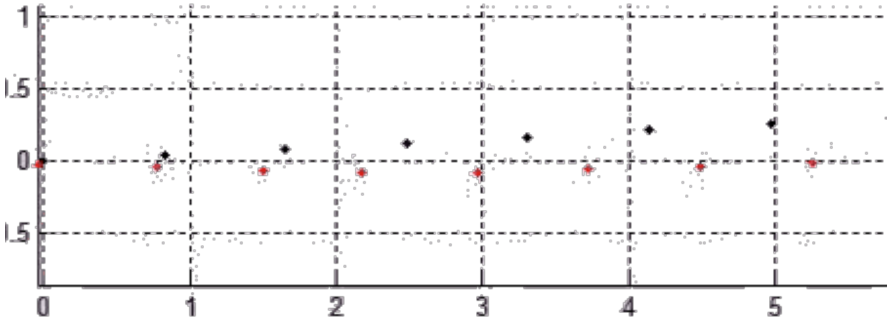





Input: Video (Image Sequence)

Result: Visual Odometry

VO



# Example transformations: perils of feature matching

I.e. what if there is image degradation?

Reference Image

Geometric transformation and blur

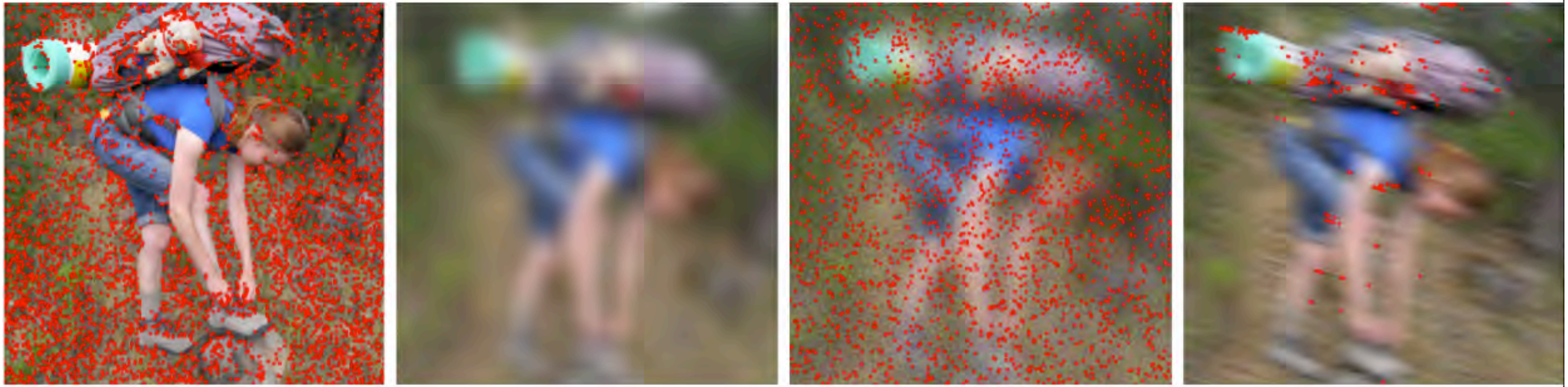
Geometric, blur and noise

Geometric, motion blur

slide courtesy of Richard Newcombe

# *Sparse* pipelines need image features

Example FAST detections (Rosten and Drummond, ECCV 2006)



Reference Image

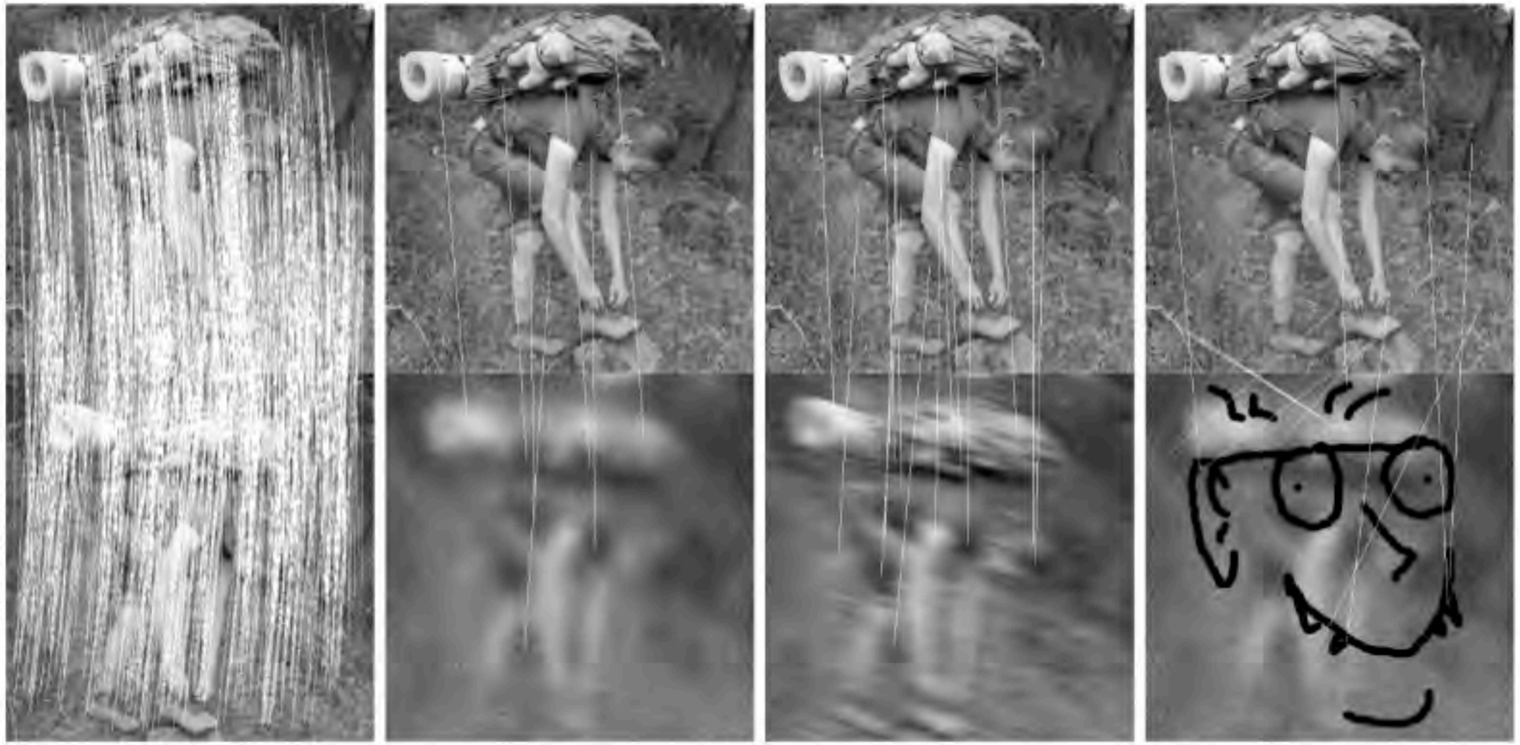
Geometric transformation and blur

Geometric, blur and noise

Geometric, motion blur

# *Sparse* (1) extraction and (2) matching

Descriptor extraction and matching using naively applied SIFT (Lowe, ICCV 2004)



Geometric only

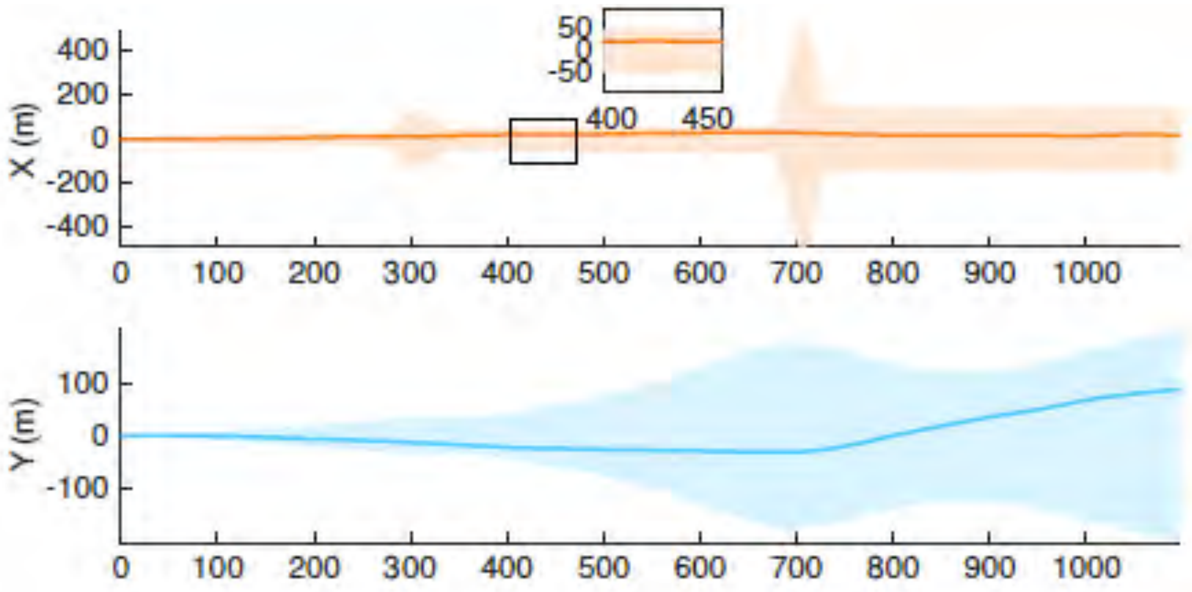
Geometric, blur and noise

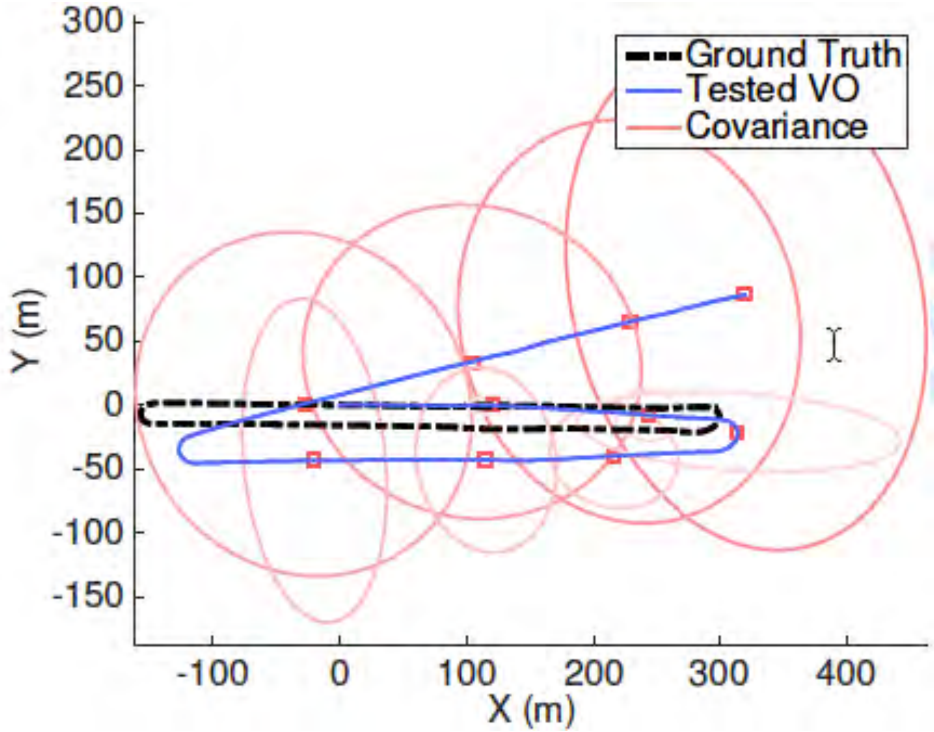
Geometric, motion blur

Geometric, blur, noise, occlusion

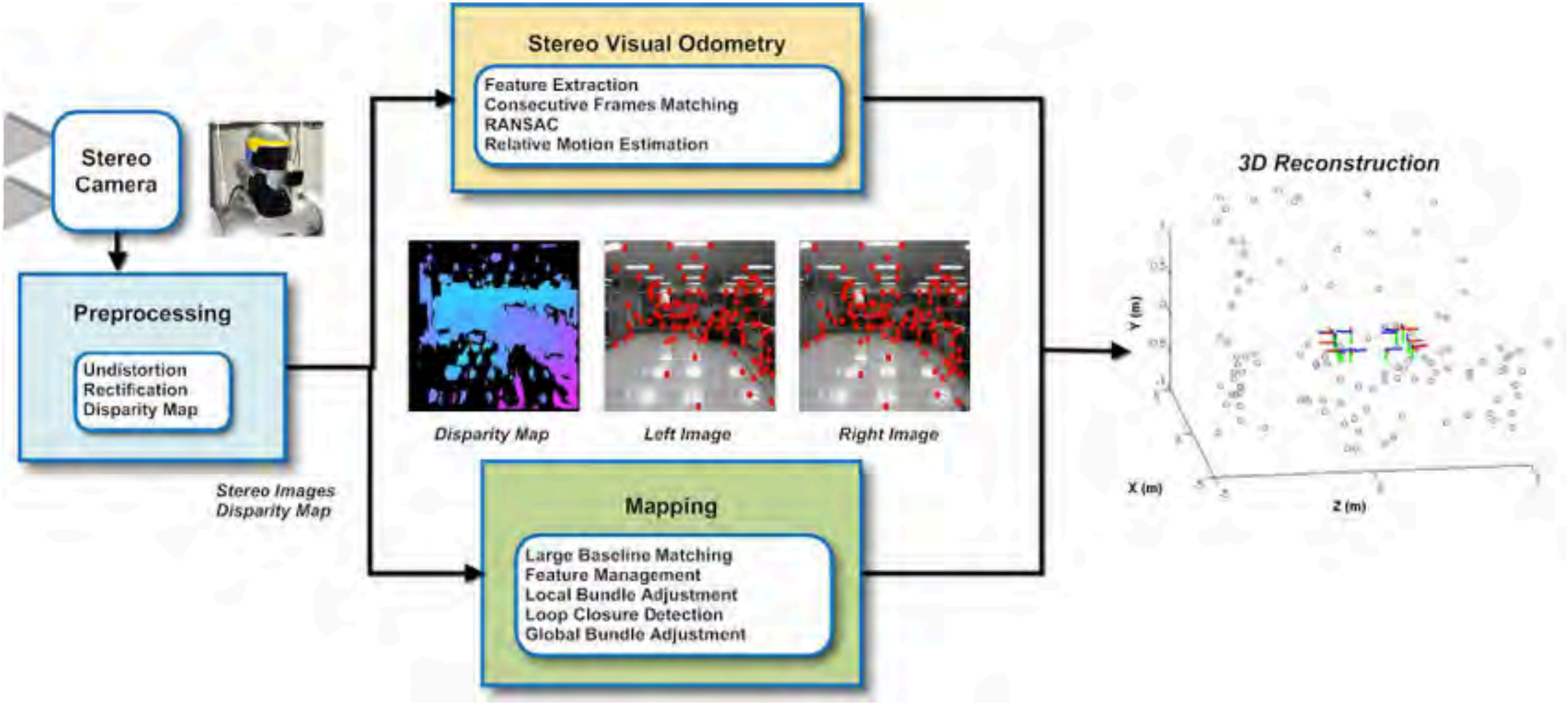
# Odometry Drift

- drift problem of VO
  - errors accumulate over time
  - similar to wheel odometry



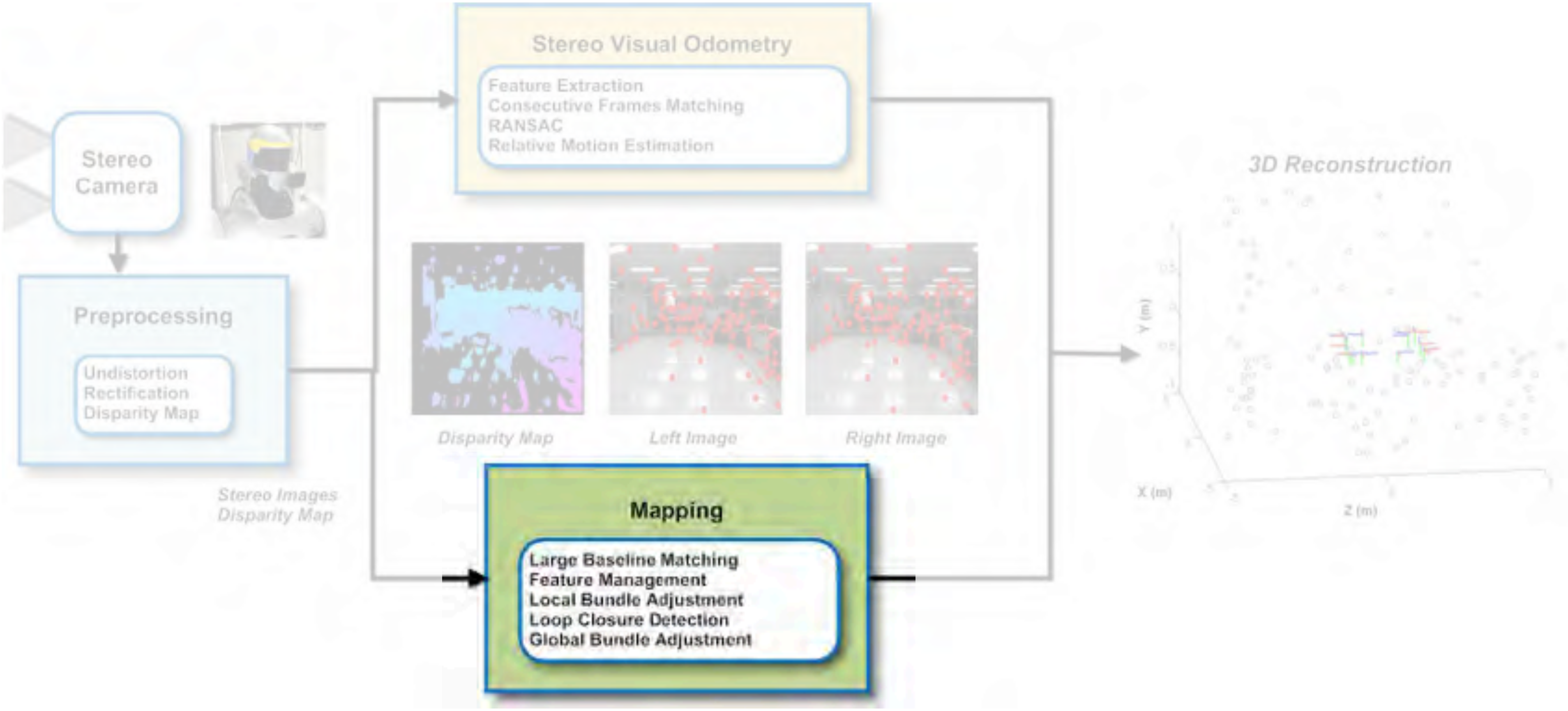



# Visual SLAM Pipeline




Alcantarilla et al. AR'13

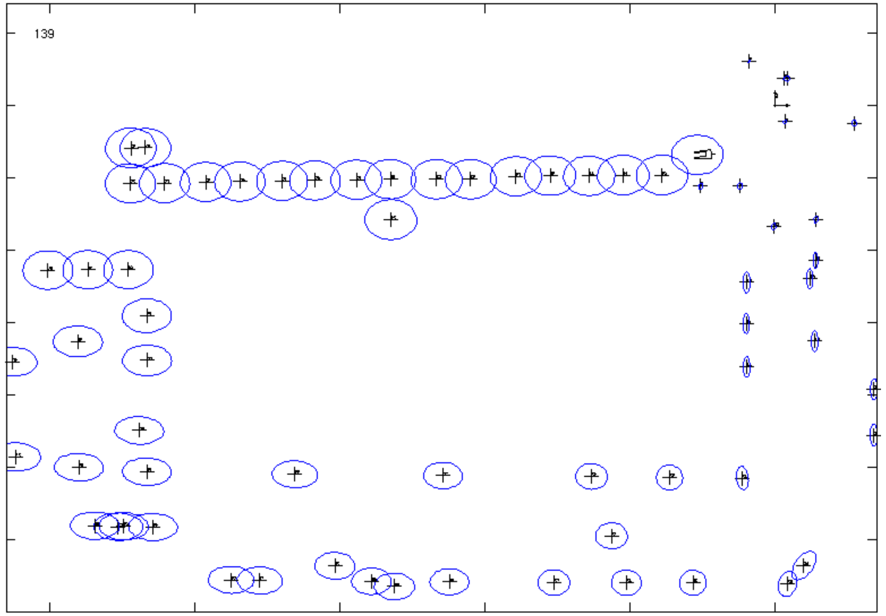
# Visual SLAM Pipeline



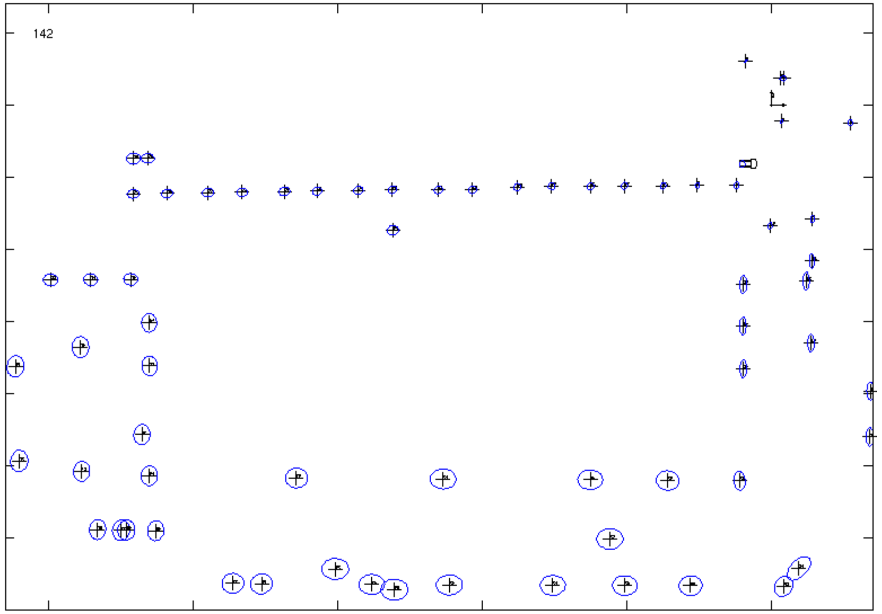

Alcantarilla et al. AR'13

# Loop Closure

- loop closure: identifying when the robot is revisiting a previous location, typically after a long exploration path (the robot "closes a loop”)
- Uncertainties **collapse** after a loop closure (whether the closure was correct or not)

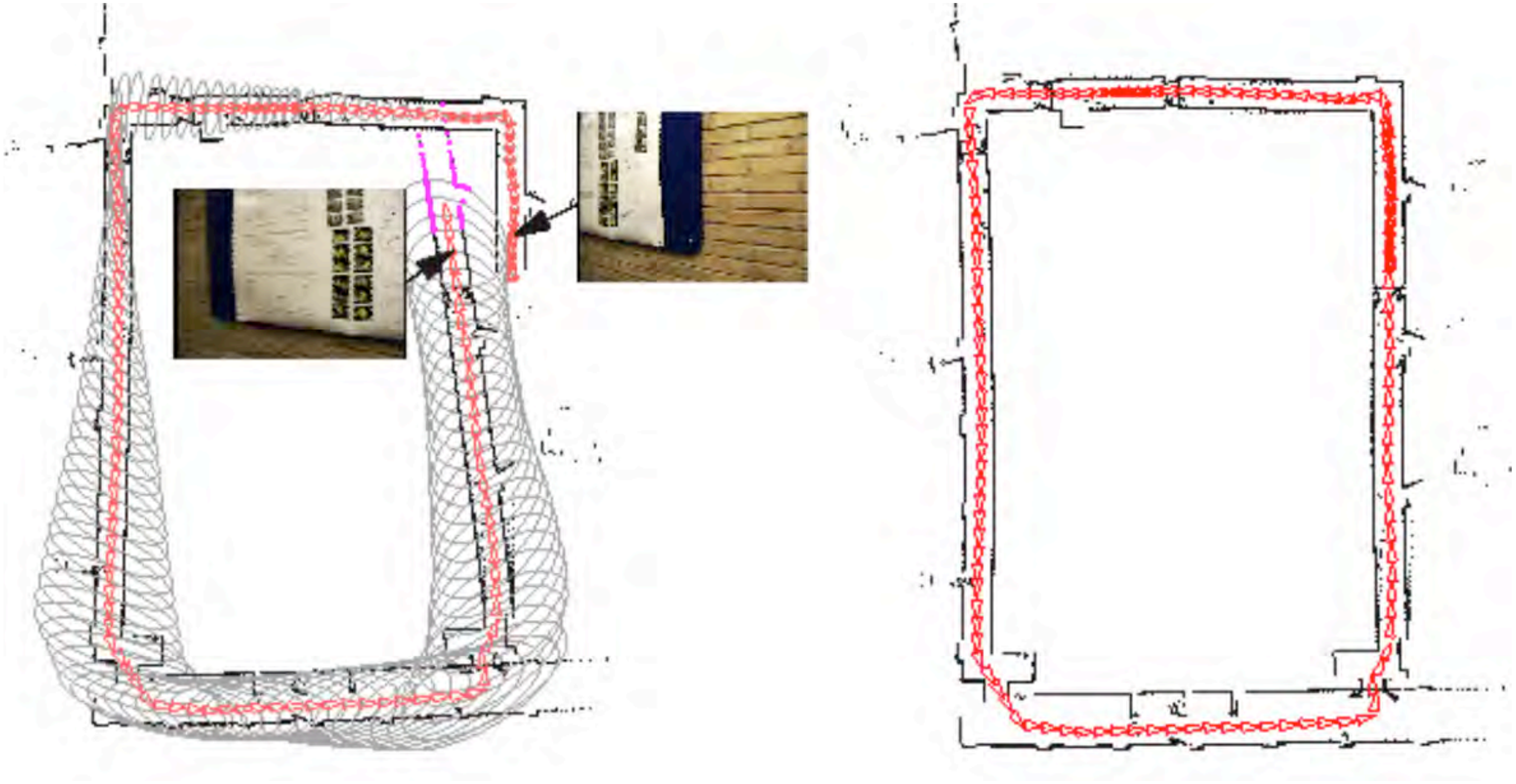



before




after

# Loop Closure Detection

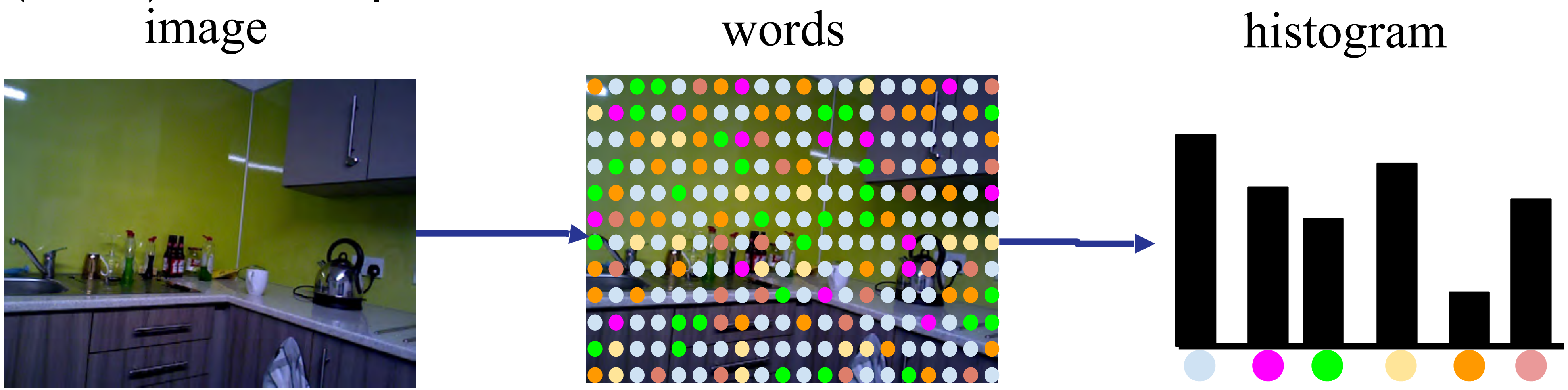


Uncertainty Update

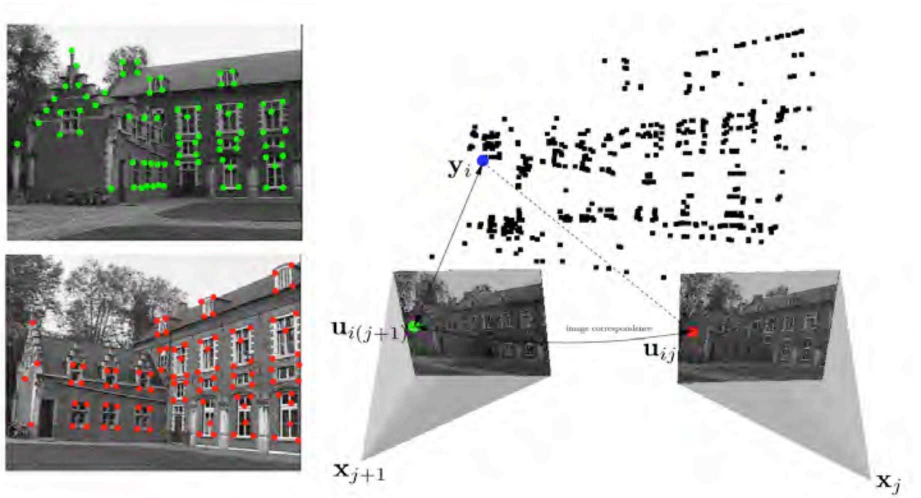
[Ho and Newman, 2006]

# Vision based Loop Closure Detection

- Place recognition: sparse local features and the Bag-of-Words (BoW) technique

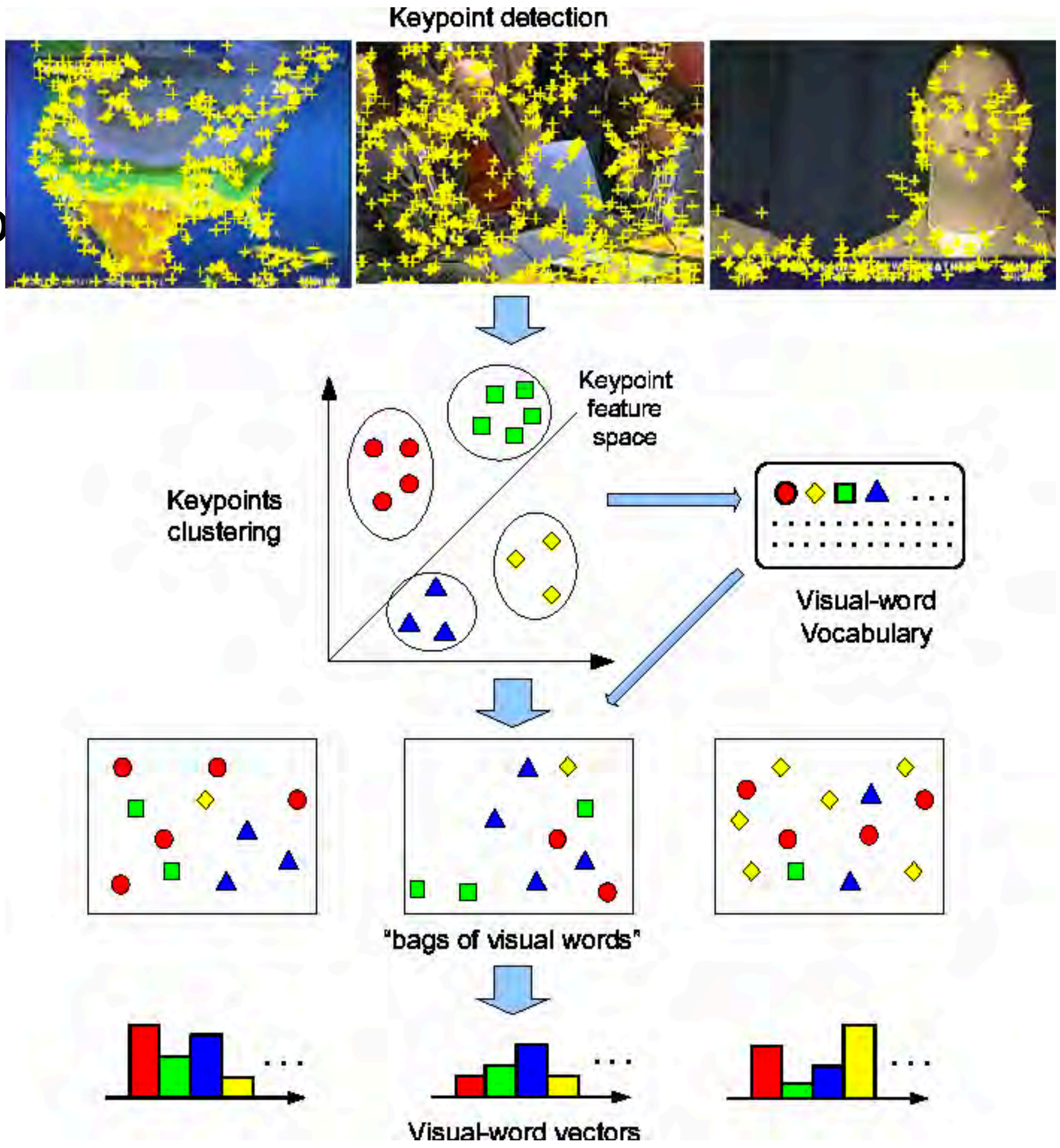



- transformation calculation



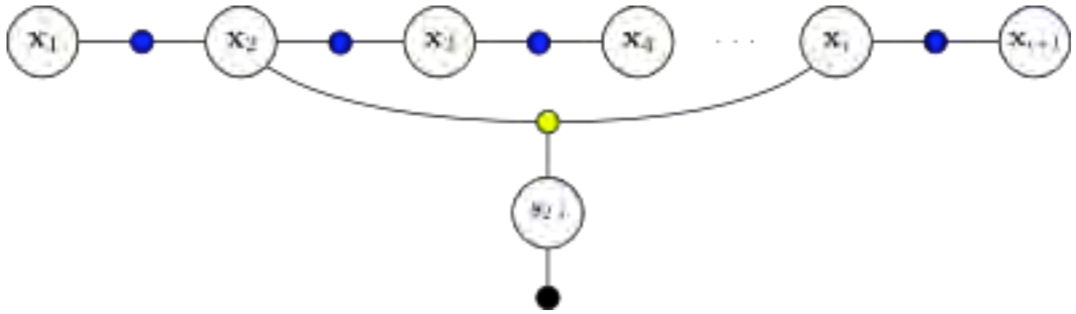
# Visual Bag-of-Words

- BoW for loop closure detectio
- place recognition



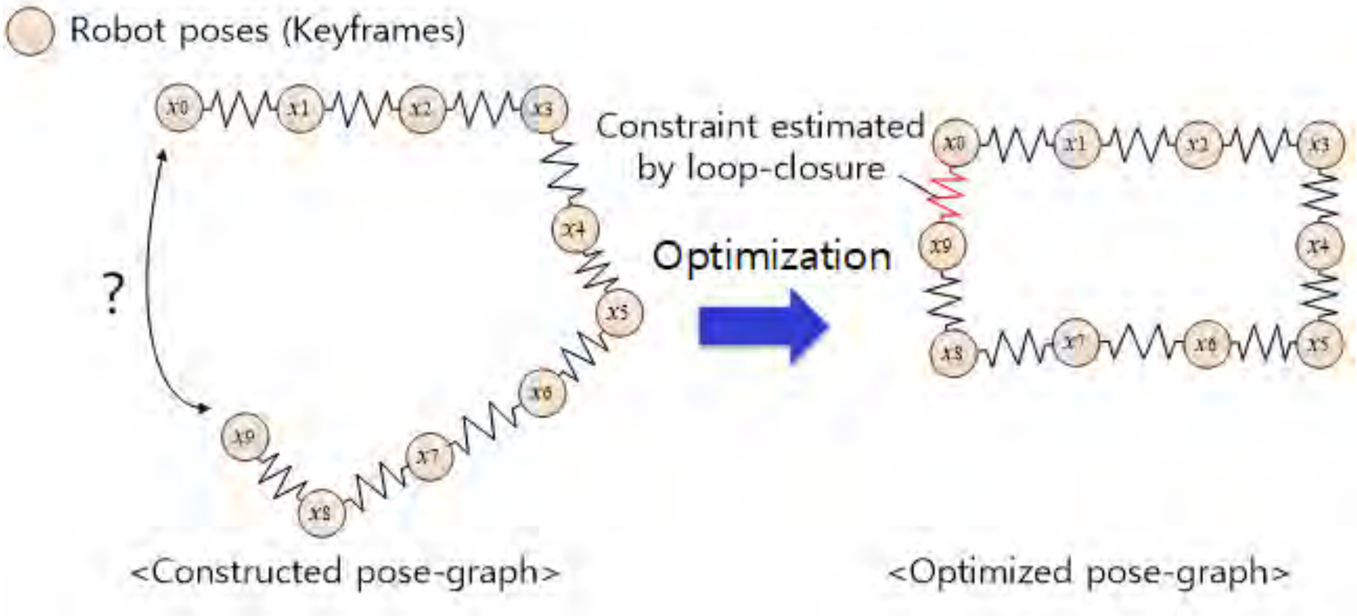
# Graph Optimization

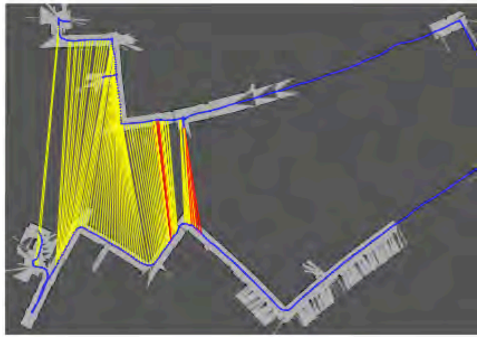
- TORO, g2o, iSAM



# Graph Optimization

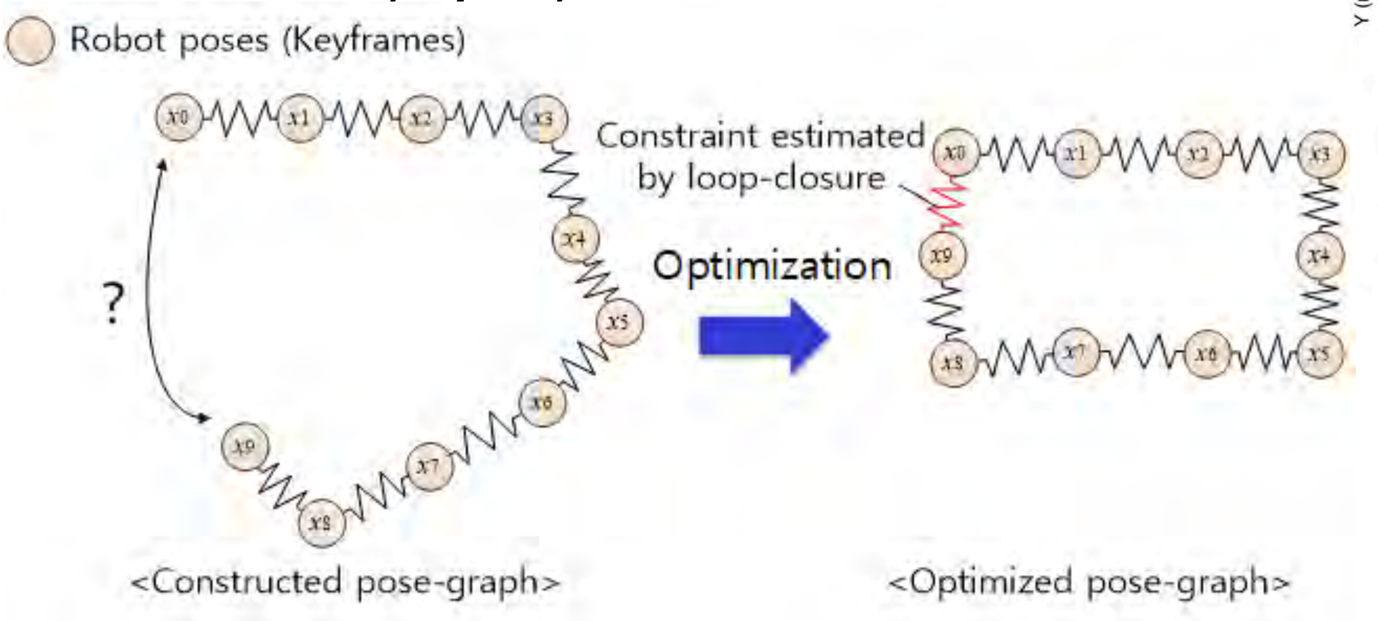
- TORO, g2o, iSAM

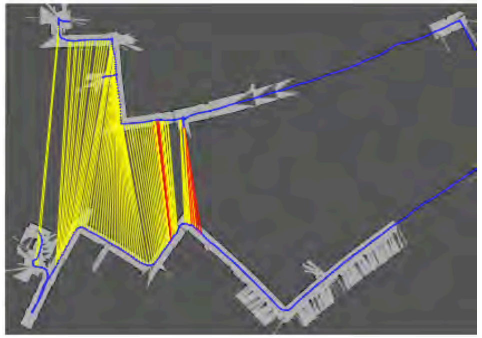


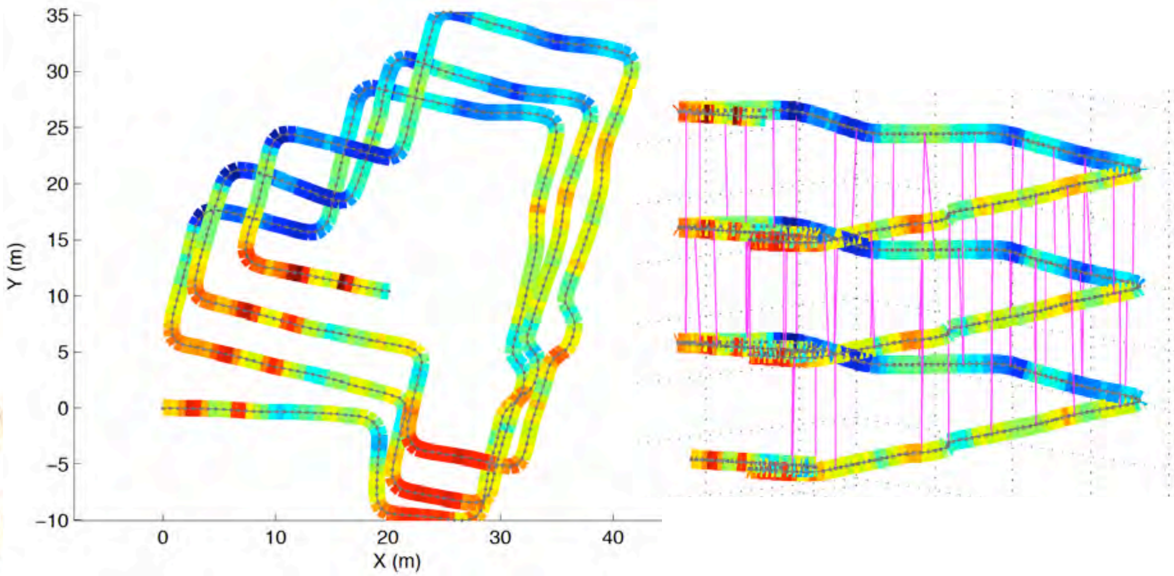



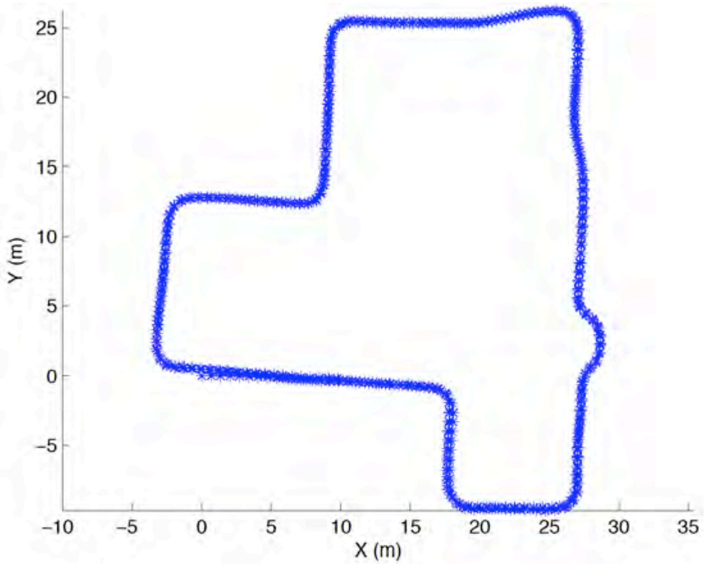
# Graph Optimization

- TORO, g2o, iSAM

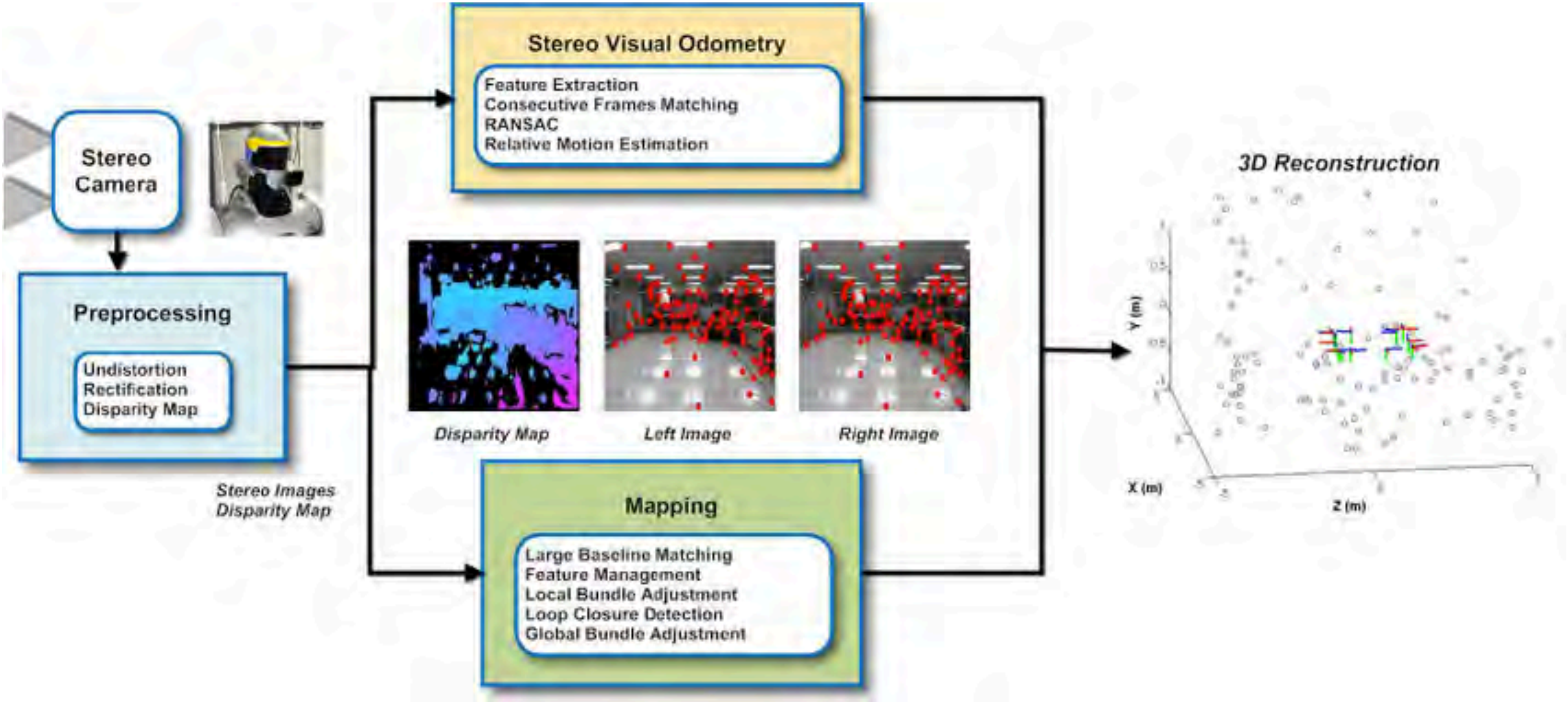


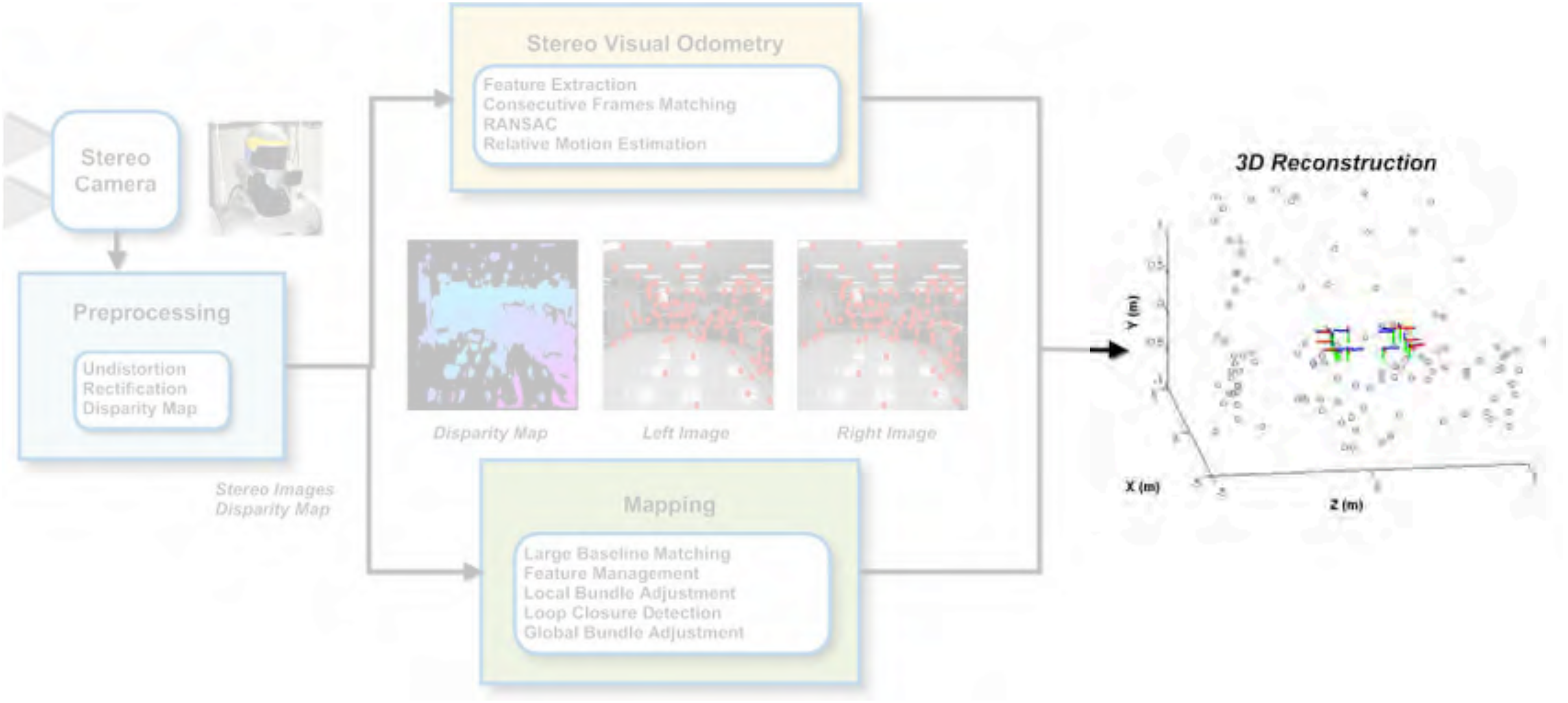



# Visual SLAM Pipeline



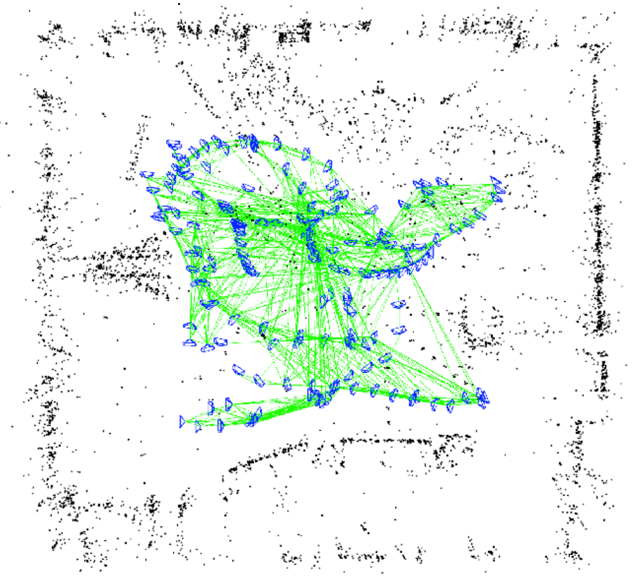

Alcantarilla et al. AR'13

# Visual SLAM Pipeline



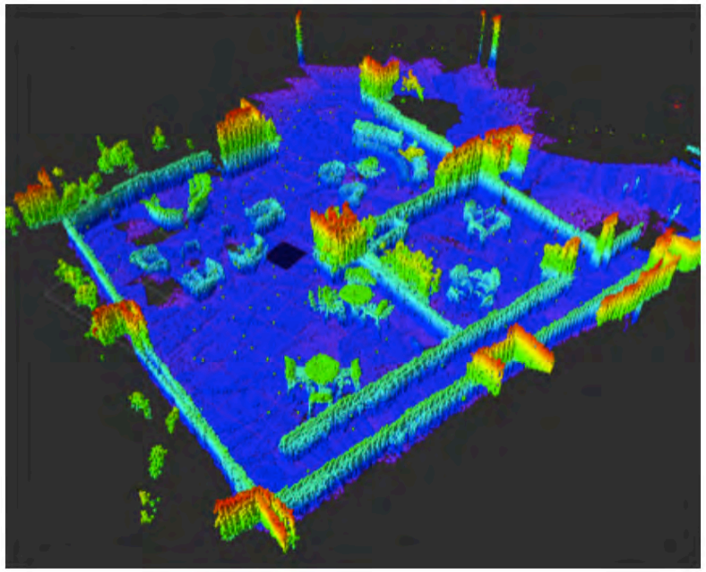

Alcantarilla et al. AR'13

# 3D Reconstruction (Map)



3D sparse feature

3D dense PCL



3D Octomap

# Video

stereo version

# Video

stereo version

# Video

stereo version

# Types

- Geometry based Methods: extract geometric constraints from imagery to estimate motion
  - **Sparse feature based methods**
  - Direct/Dense methods

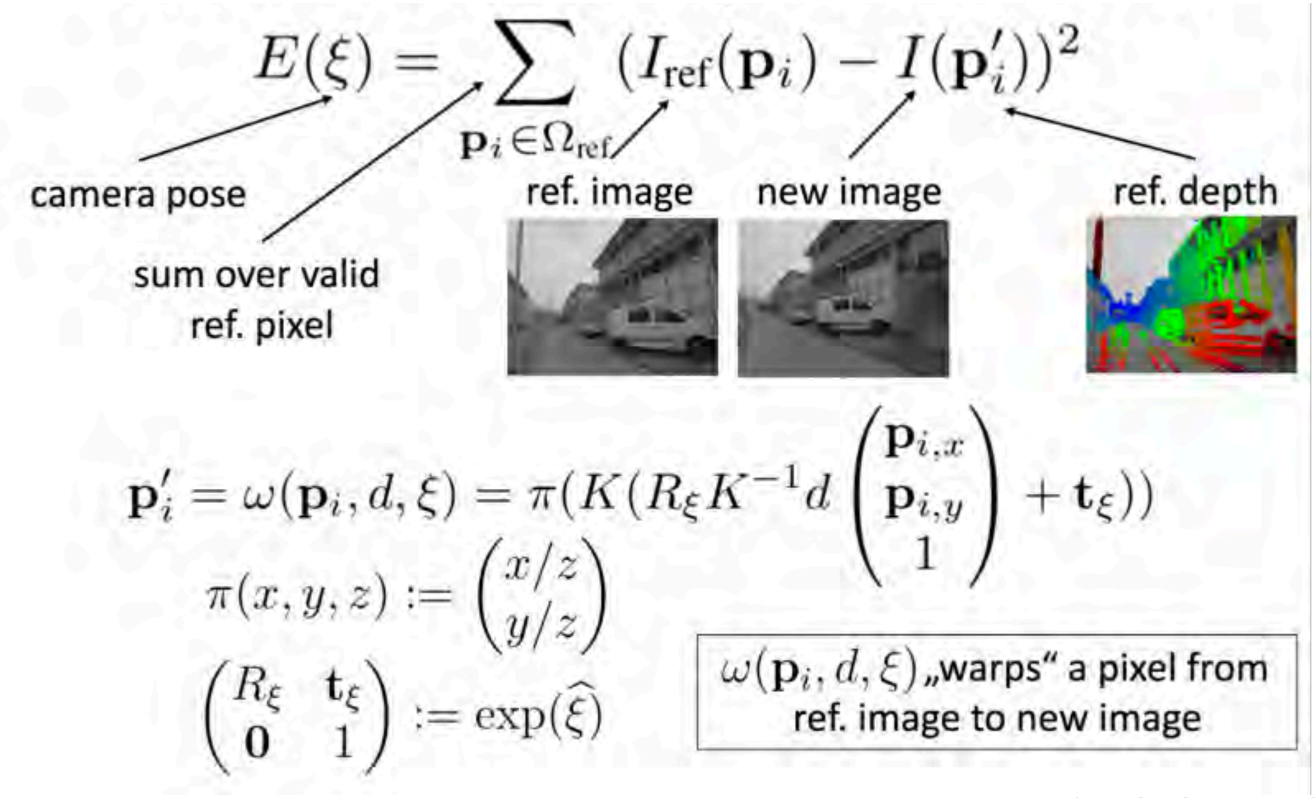
- Learning based Methods: use machine learning techniques without explicitly applying geometric theory

# Direct methods

- Direct methods: **find camera poses** that maximise the probability of observing current **image** (pixel intensities)

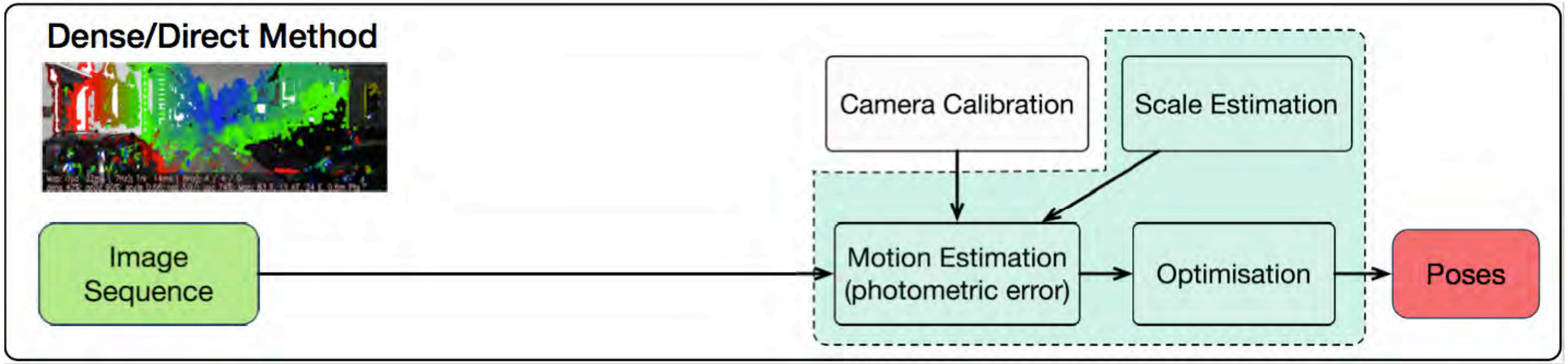
- whole image rather than sparse features

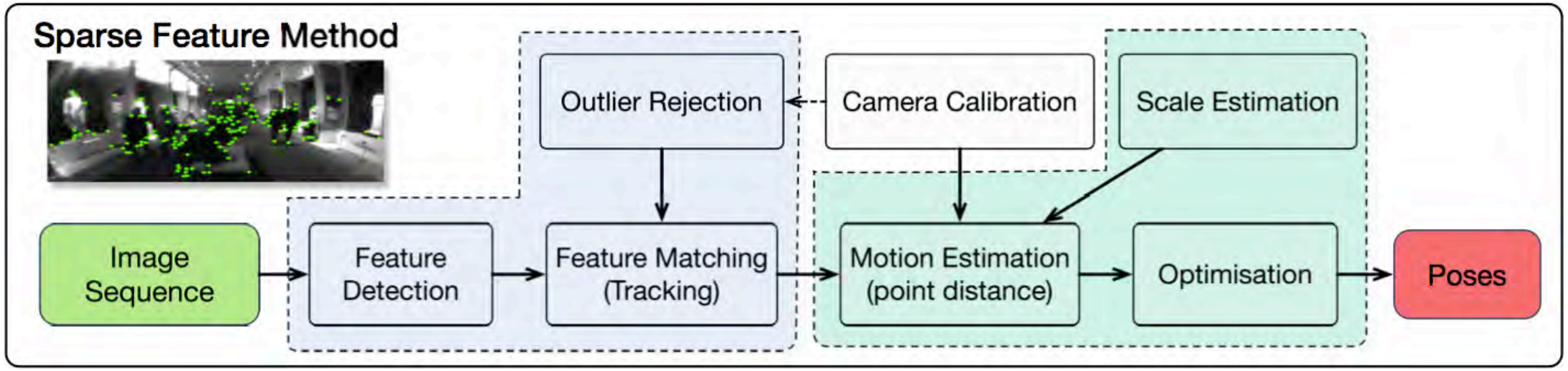
- assumption: **brightness constancy**
- minimise photometric error



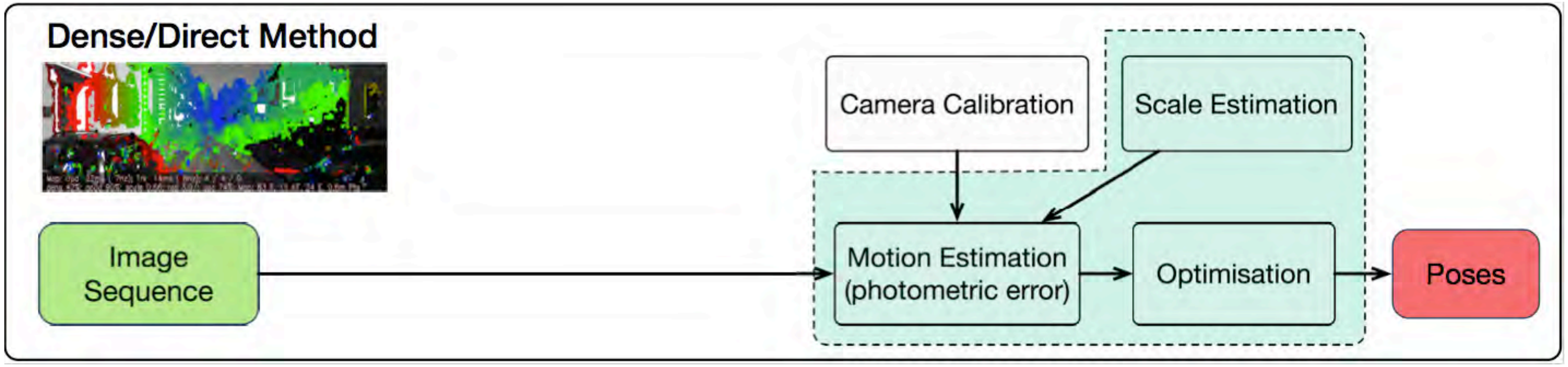

courtesy of Jakob Engel

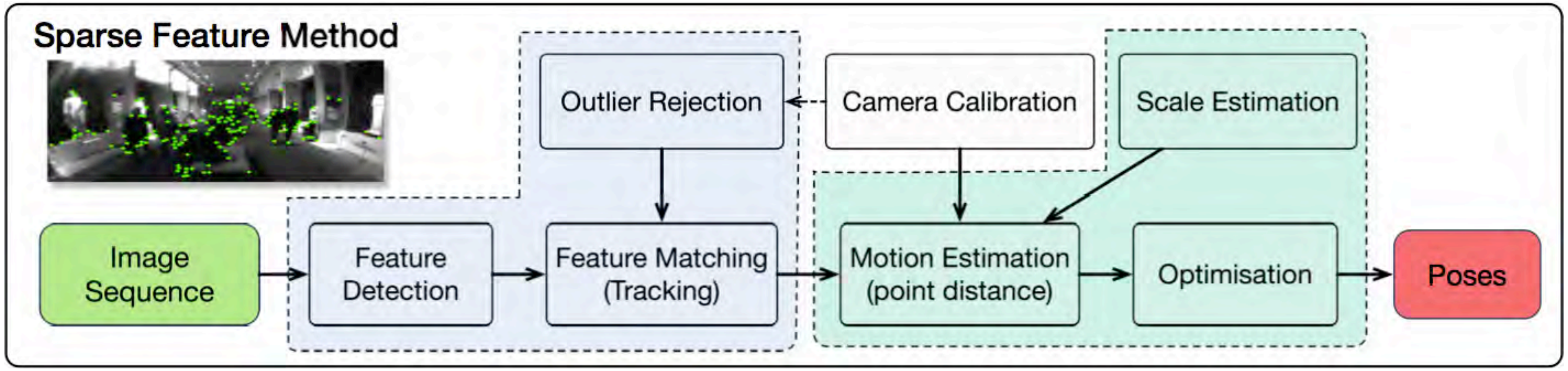
# Direct methods



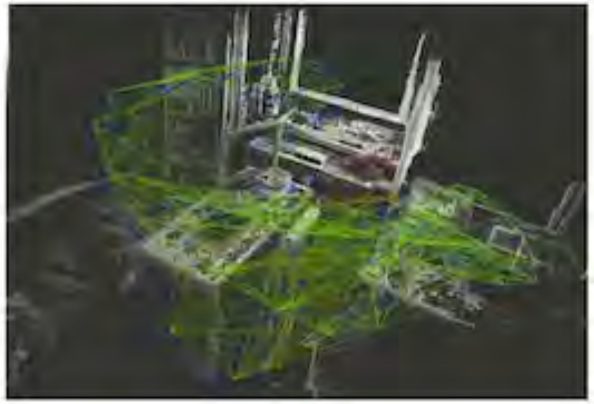



# Direct methods



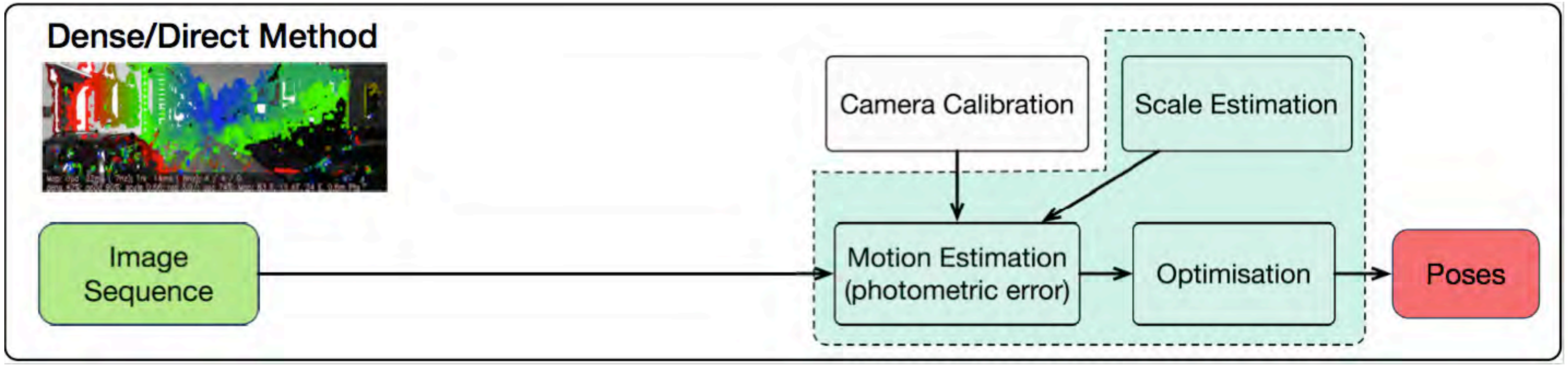

Direct SLAM

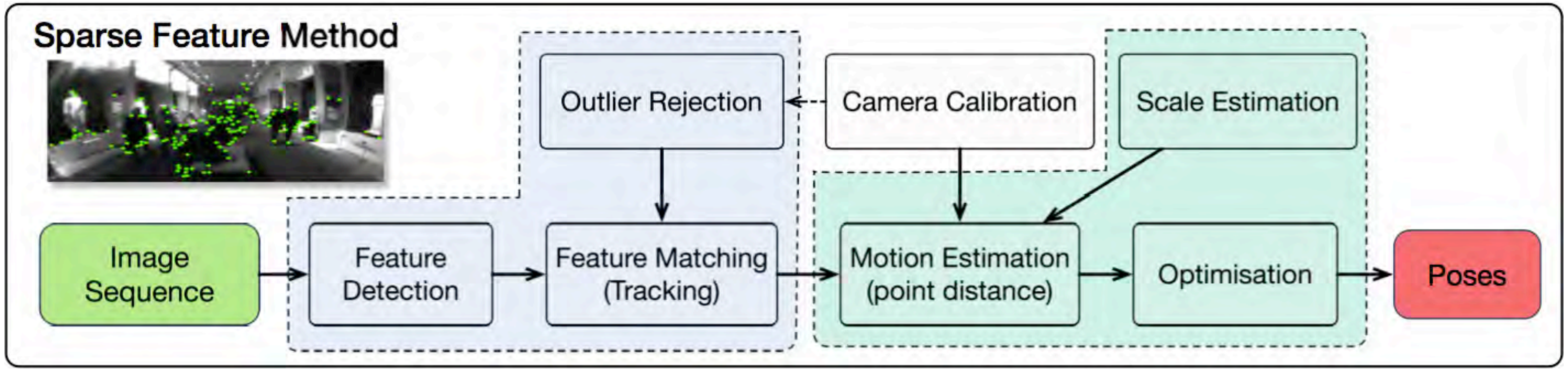


Minimize **Photometric Error**

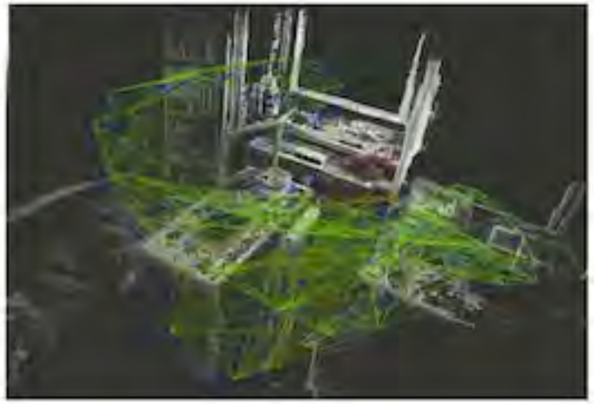
**Semi Dense / Dense** Reconstruction

# Direct methods



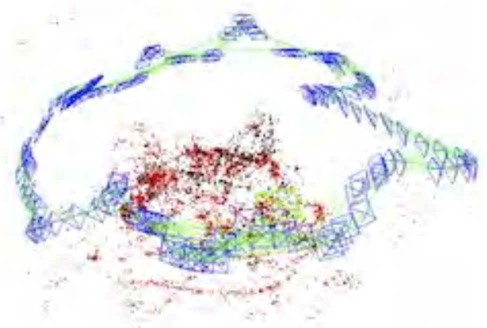

**Direct SLAM**



Minimize **Photometric Error**

**Semi Dense / Dense** Reconstruction

**Feature-Based SLAM**

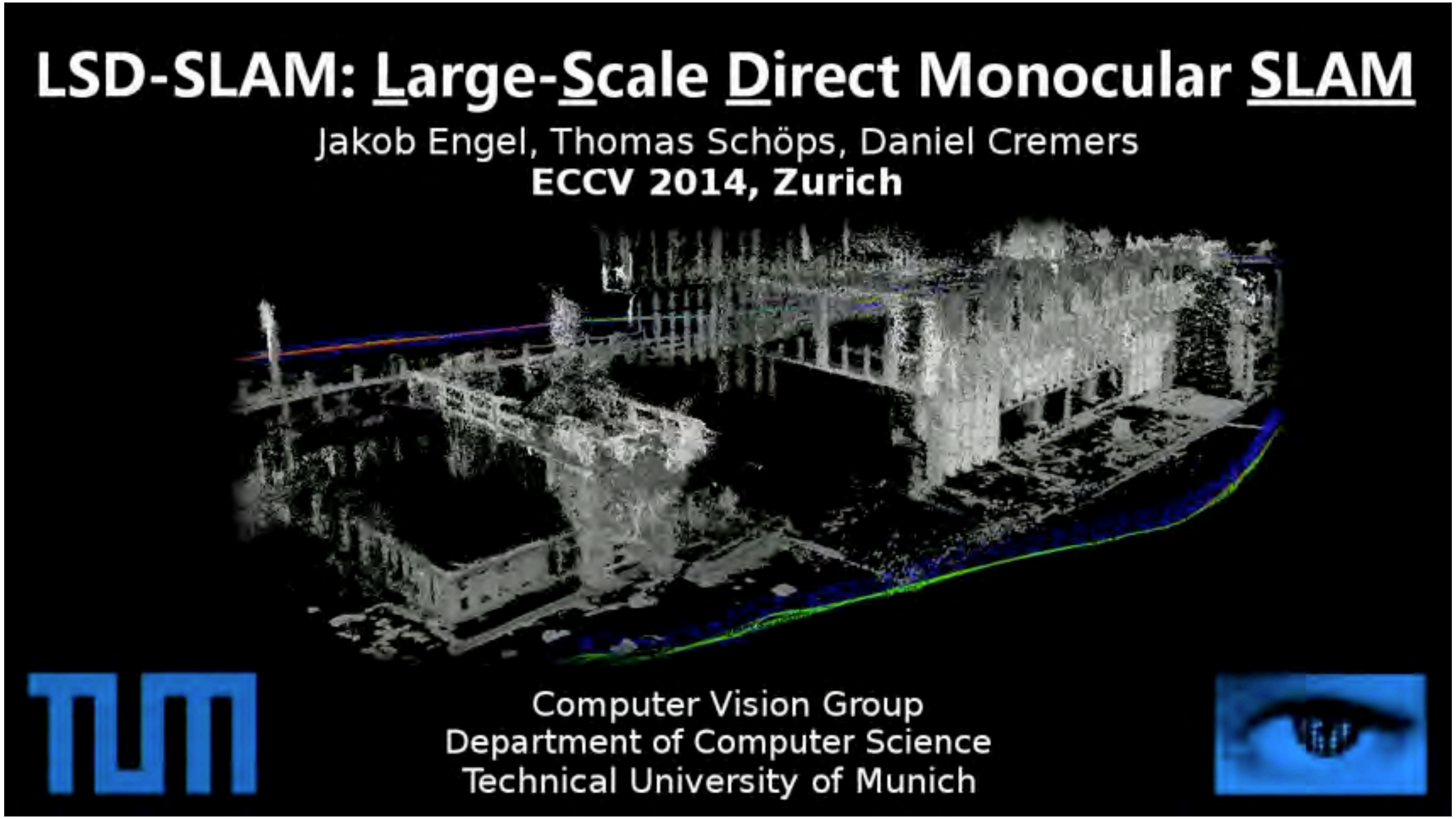


Minimize **Feature Reprojection Error**

**Sparse** Reconstruction

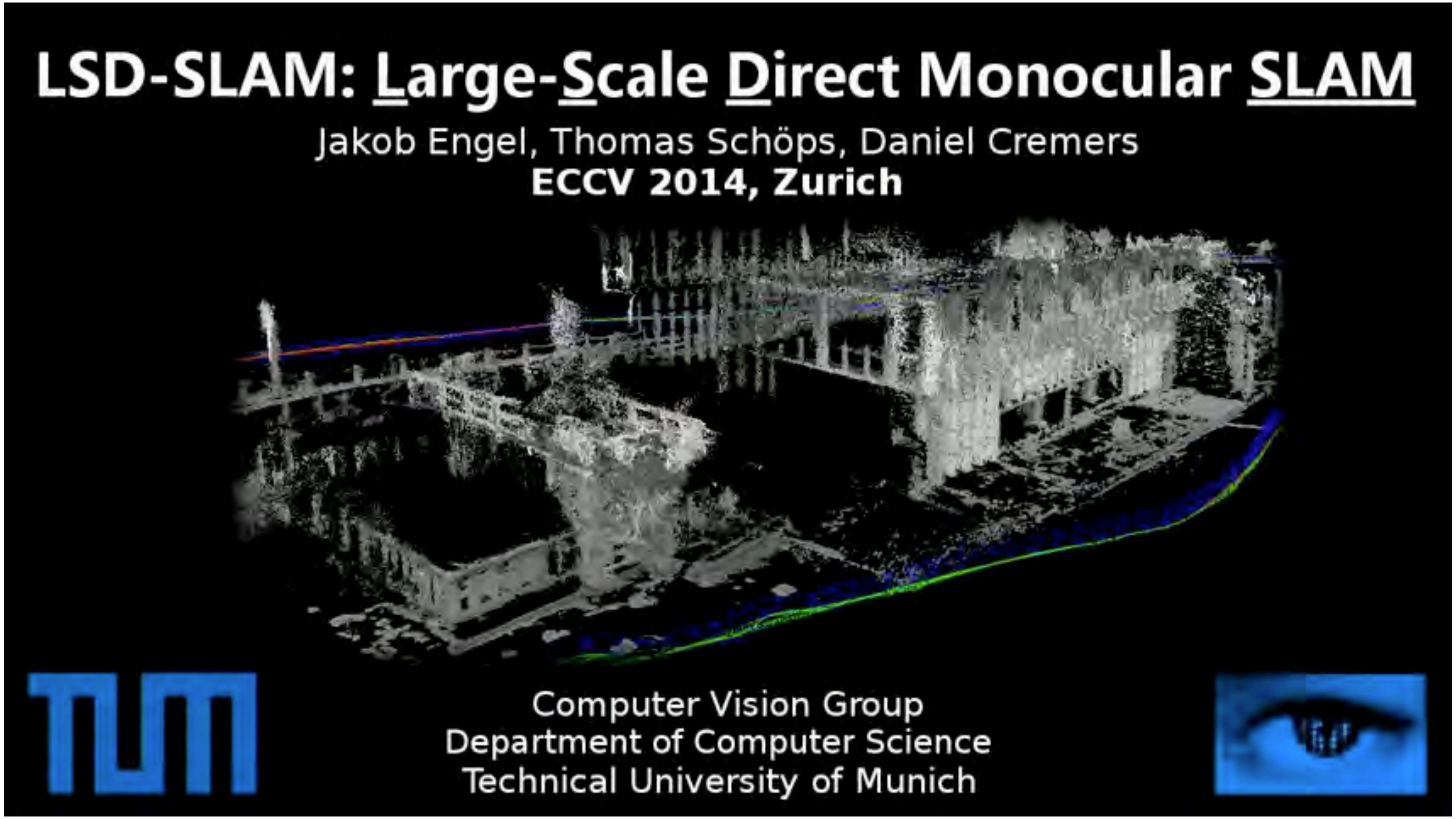
# Video

Dense method



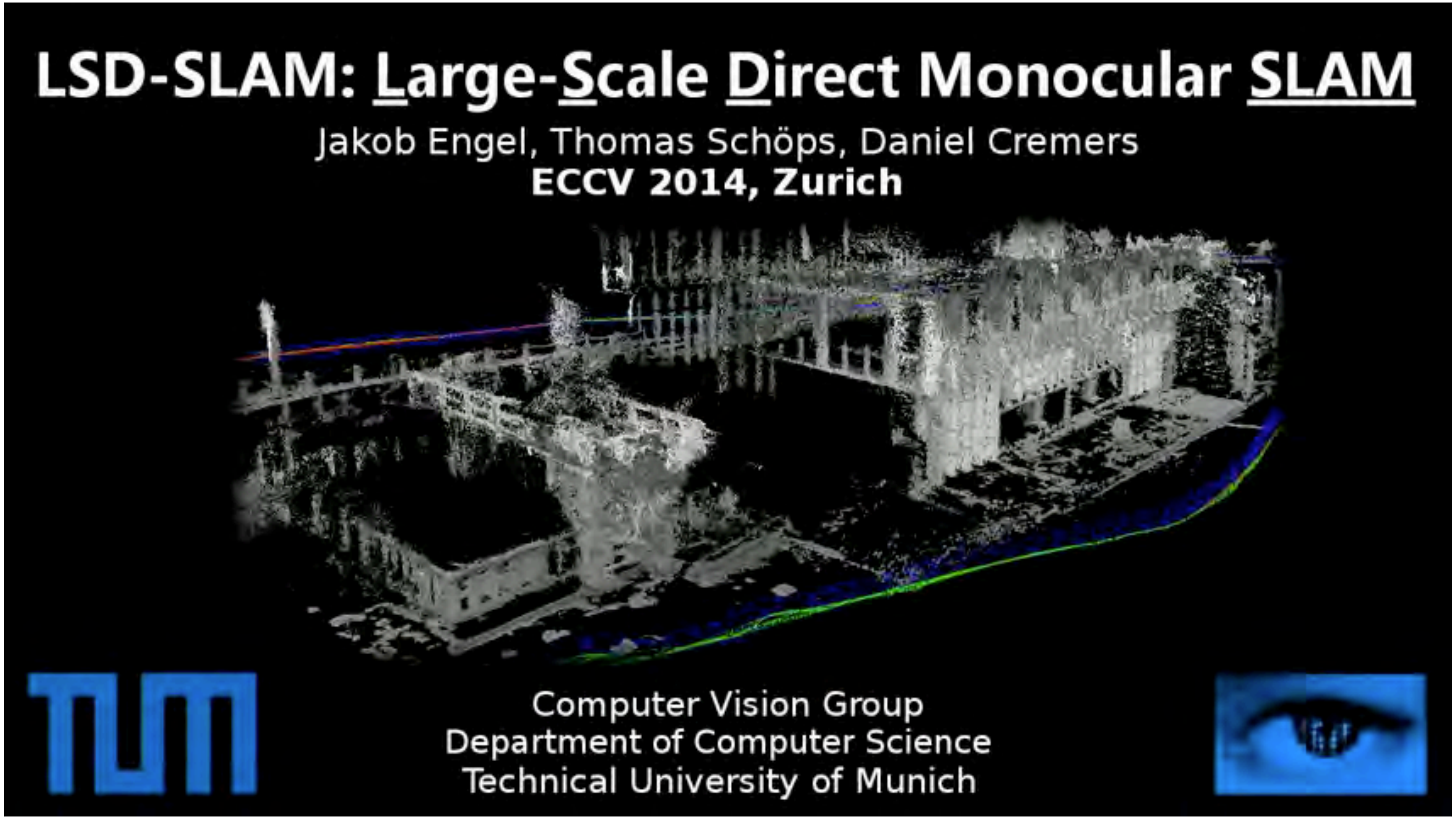
# Video

Dense method



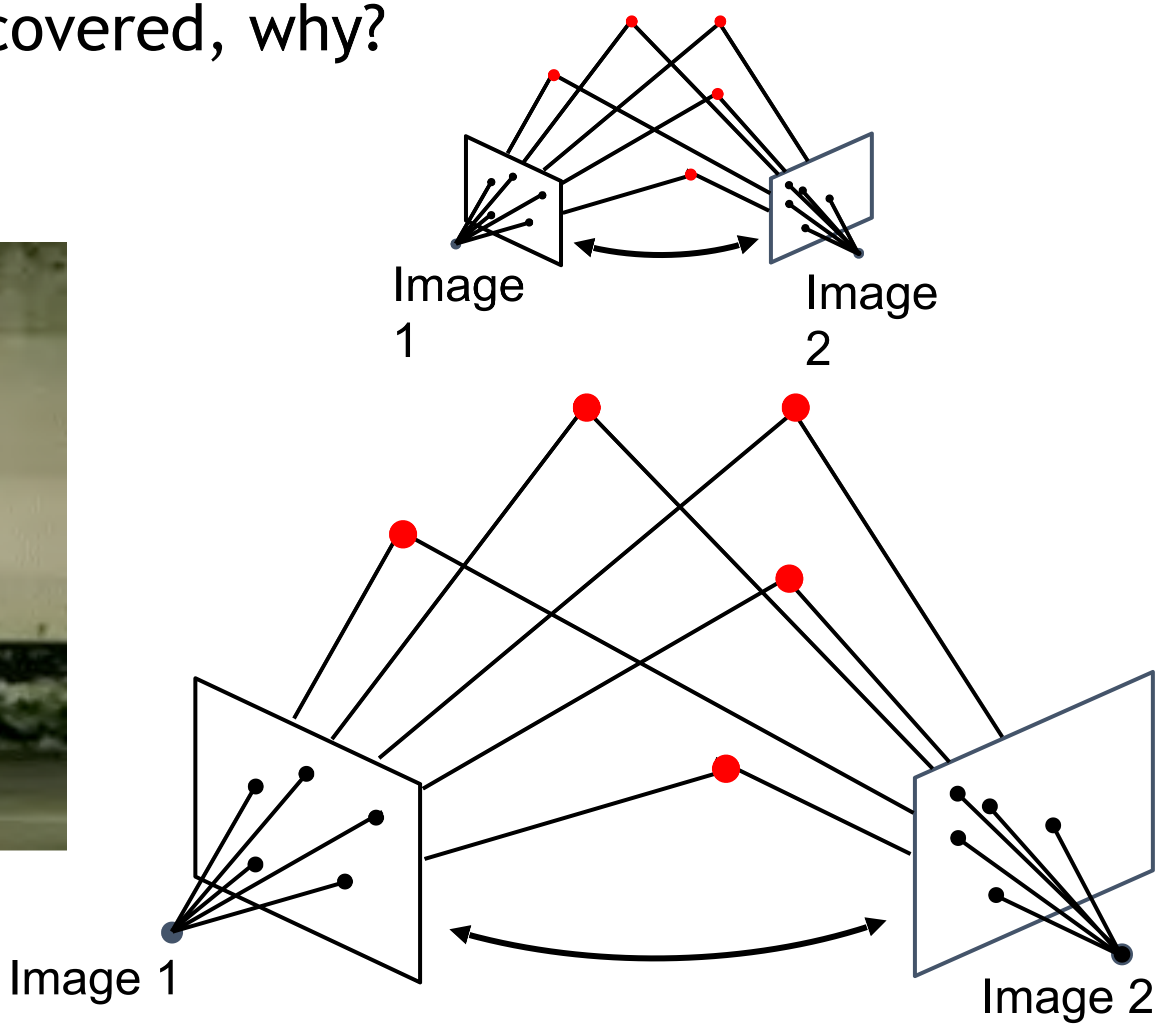
# Video

Dense method



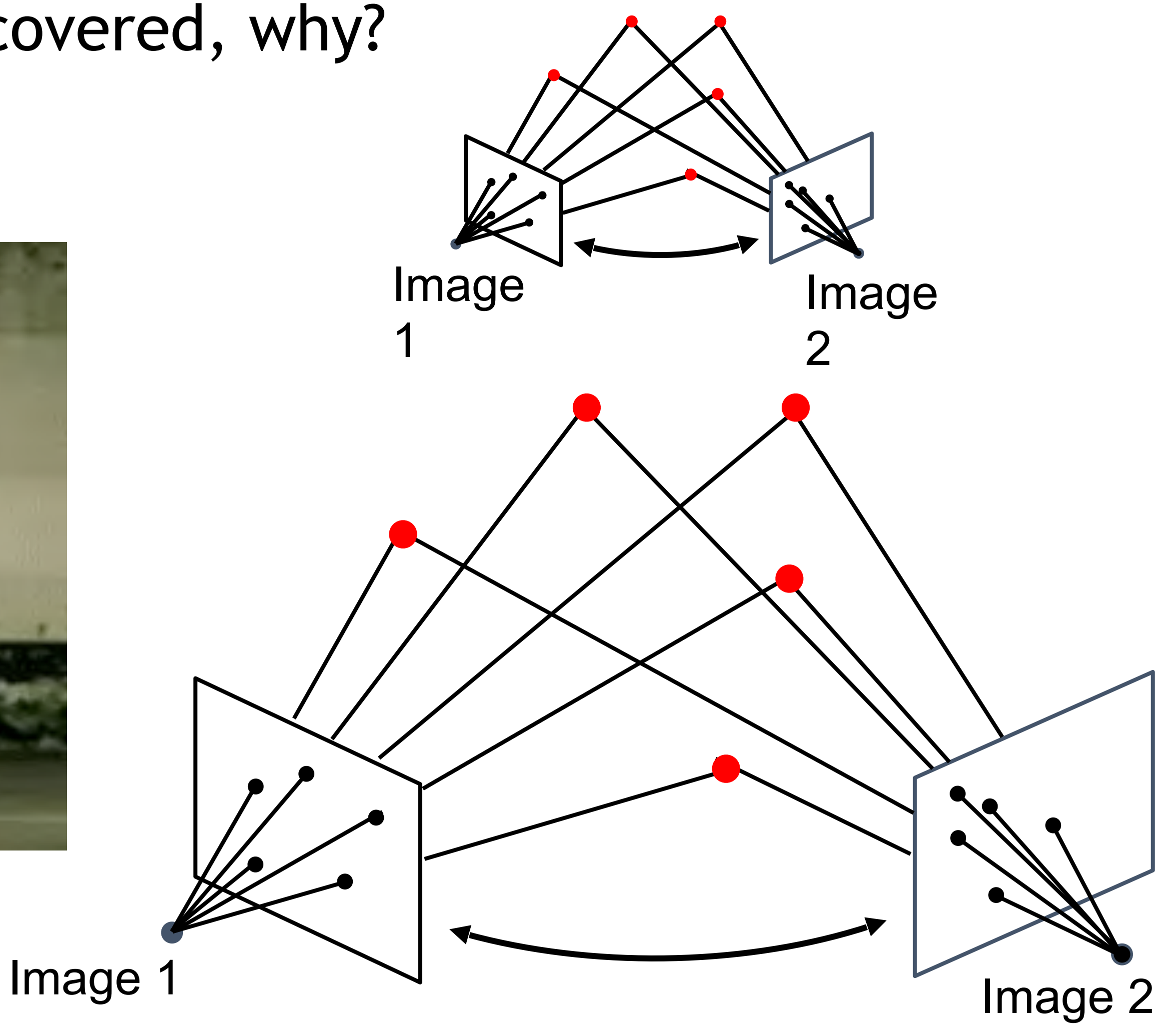
# Scale Problem of Monocular VO

- metric scale cannot be recovered, why?



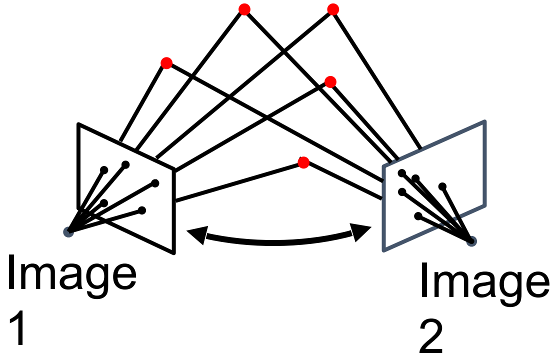
# Scale Problem of Monocular VO

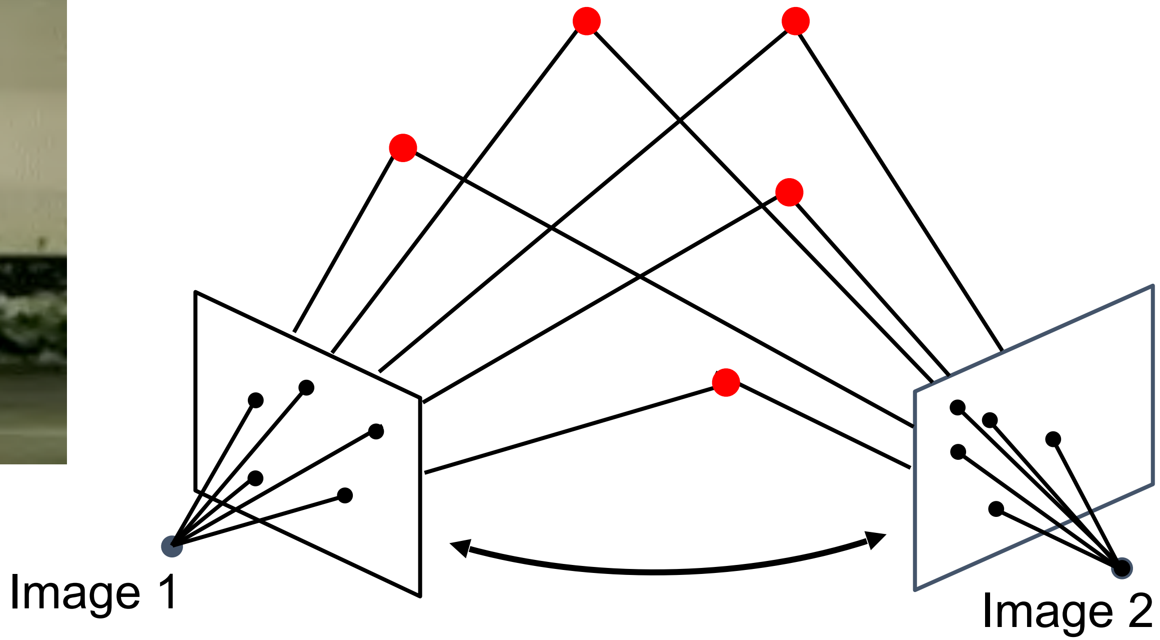
- metric scale cannot be recovered, why?



# Scale Problem of Monocular VO

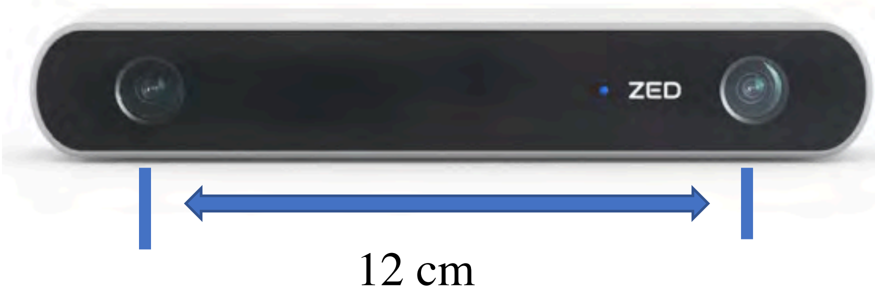
- metric scale cannot be recovered, why?



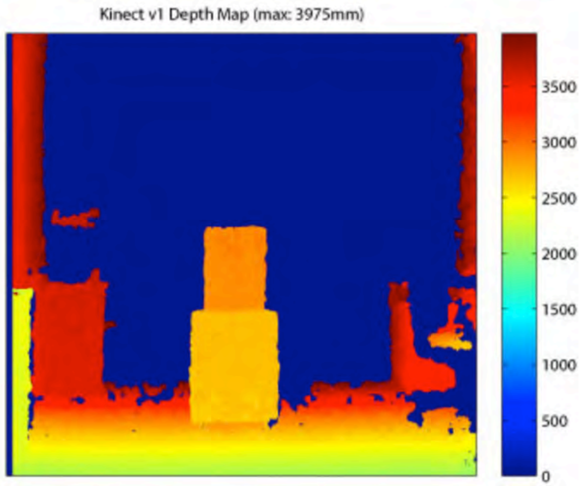



# Recover Absolute Depth

- Stereo or RGB-D sensors





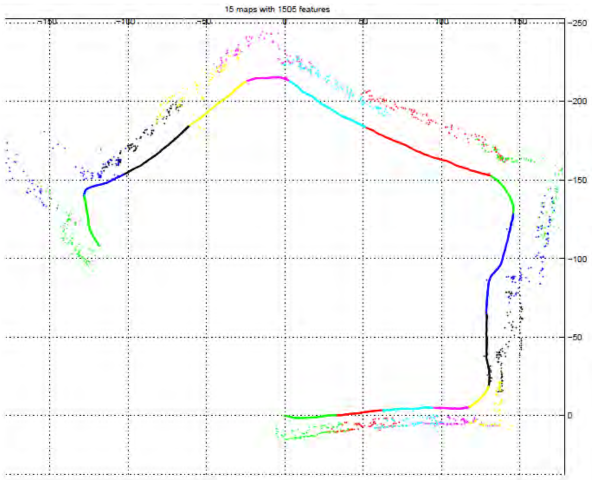



- Fuse with inertial sensor (Visual Inertial Odometry/SLAM) or prior knowledge

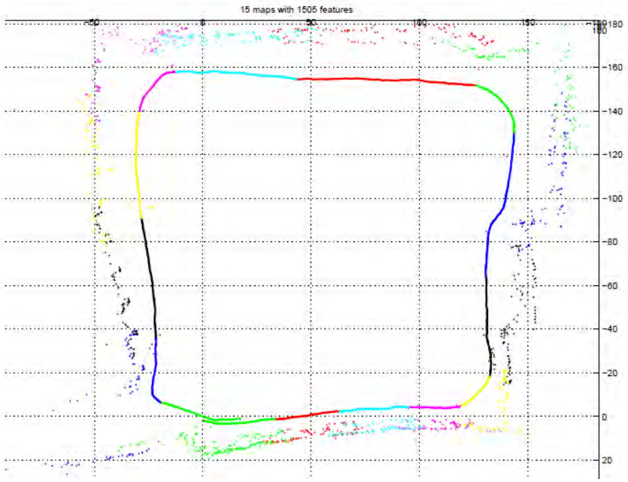
# VO vs. Visual SLAM (1/2)

- The goal of SLAM in general is to obtain a global, consistent estimate of the robot path. This is done through identifying loop closures. When a loop closure is detected, this information is used to reduce the drift in both the map and camera path (global bundle adjustment).

- Conversely, VO aims at recovering the path incrementally, pose after pose, and potentially optimizing only over the last $m$ poses path (windowed bundle adjustment)




**Before loop closing**

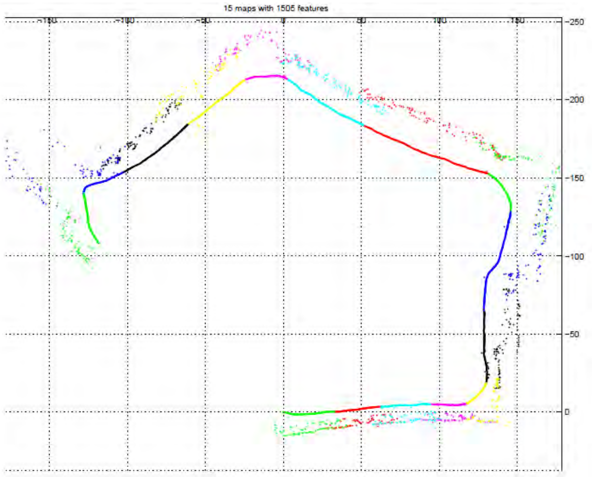



**After loop closing**

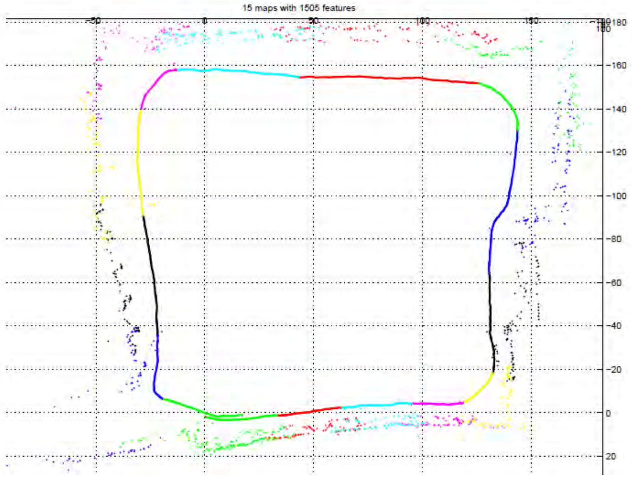
Image courtesy of Clemente et al. RSS'07

# VO vs. Visual SLAM (2/2)

- VO only aims to the local consistency of the trajectory
- SLAM aims to the global consistency of the trajectory and of the map
- VO can be used as a building block of visual SLAM
- VO is visual SLAM before closing the loop!
- The choice between VO and V-SLAM depends on the tradeoff between performance and consistency, and simplicity in implementation.
- VO trades off consistency for real-time performance, without the need to keep track of all the previous history of the camera.




**Visual odometry**




**Visual SLAM**

Image courtesy of Clemente et al. RSS'07

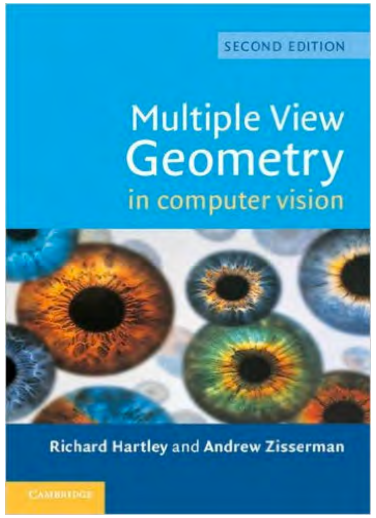
# Good for VO/SLAM?

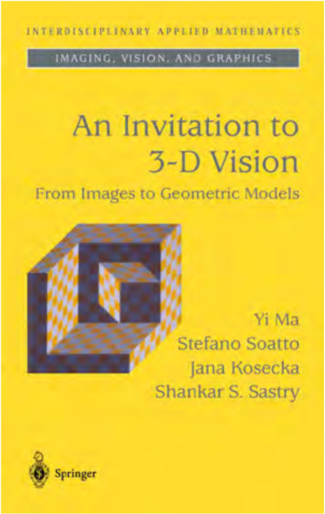
# More on Geometry based VO/SLAM

- books
- papers on Vision
- open-source algorithms to try
  - LIBVISO2
  - MonoSLAM
  - PTAM
  - ORB-SLAM

  - LSD-SLAM
  - SVO: Semi-direct Visual Odometry
  - DSO: Direct Sparse Odometry

  - RGB-D SLAM
  - RTAB-Map
  - ……….



# Types

- Geometry based Methods: extract geometric constraints from imagery to estimate motion
  - Sparse feature based methods
  - Direct/Dense methods

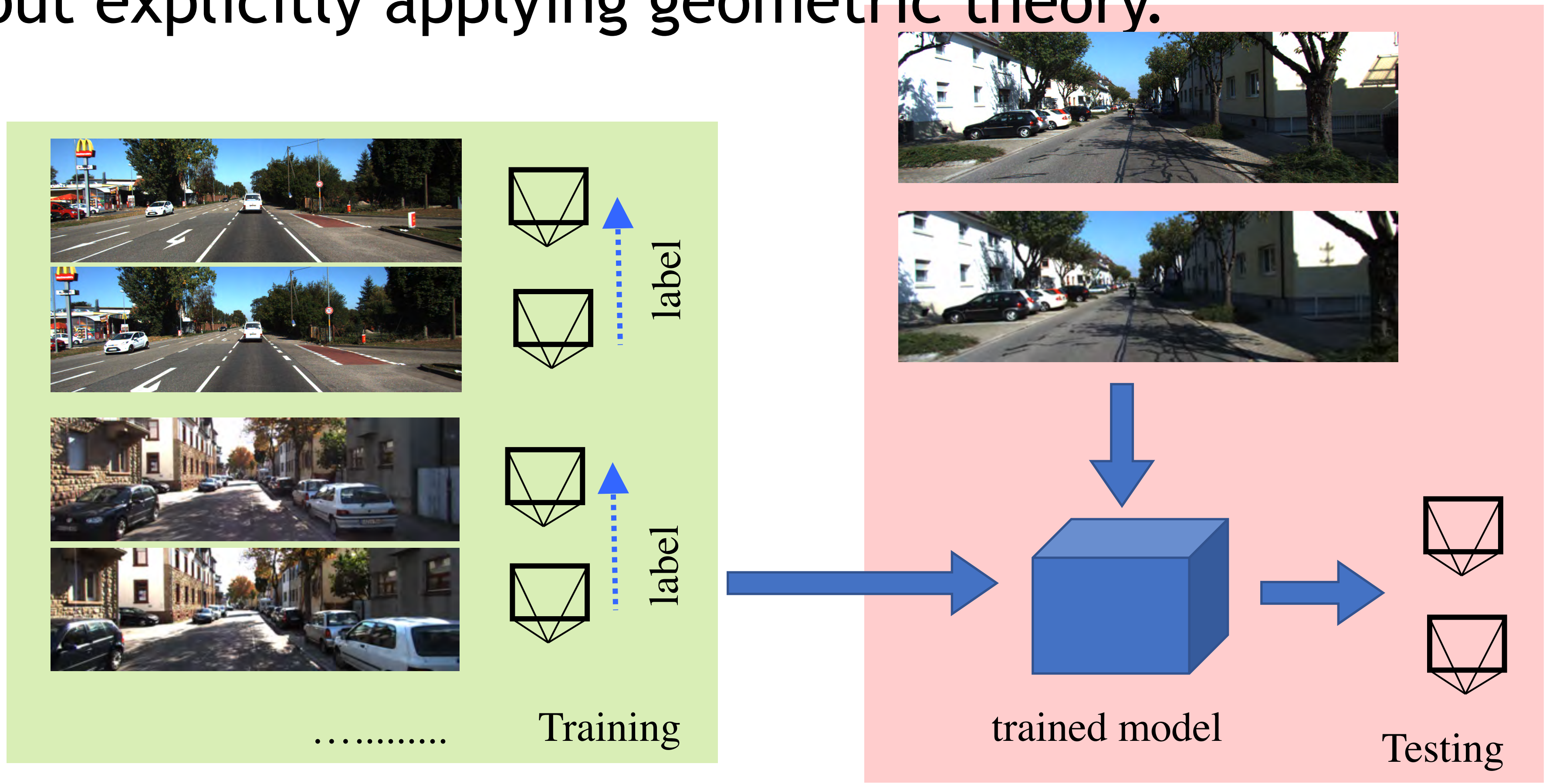
several decades
excellent accuracy
very popular

- Learning based Methods

several years
rare
increasingly popular

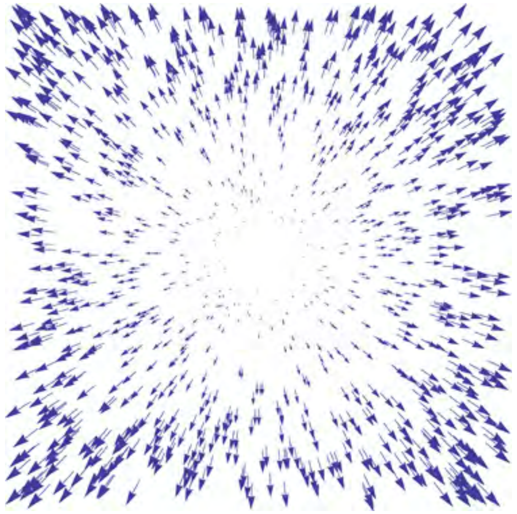
# Learning based Methods

- Learning based methods aim to derive motion model and infer VO from sensor data by using machine learning techniques without explicitly applying geometric theory.



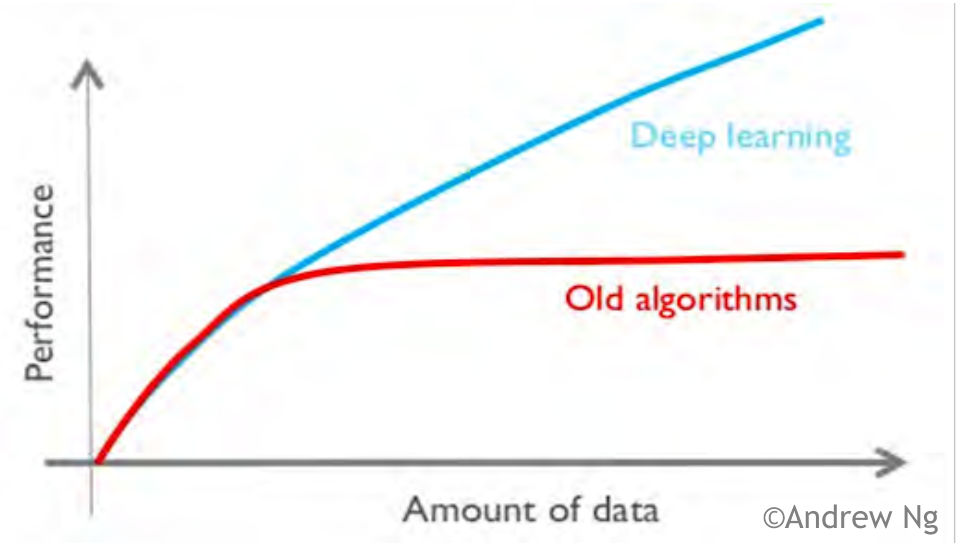
# Traditional learning based methods

- Sparse optical flow
  - K Nearest Neighbour (KNN)
  - Gaussian Processes (GP)
  - Support Vector Machines (SVM) regression
- difficult to use raw images
  - high-dimensional
  - redundant information



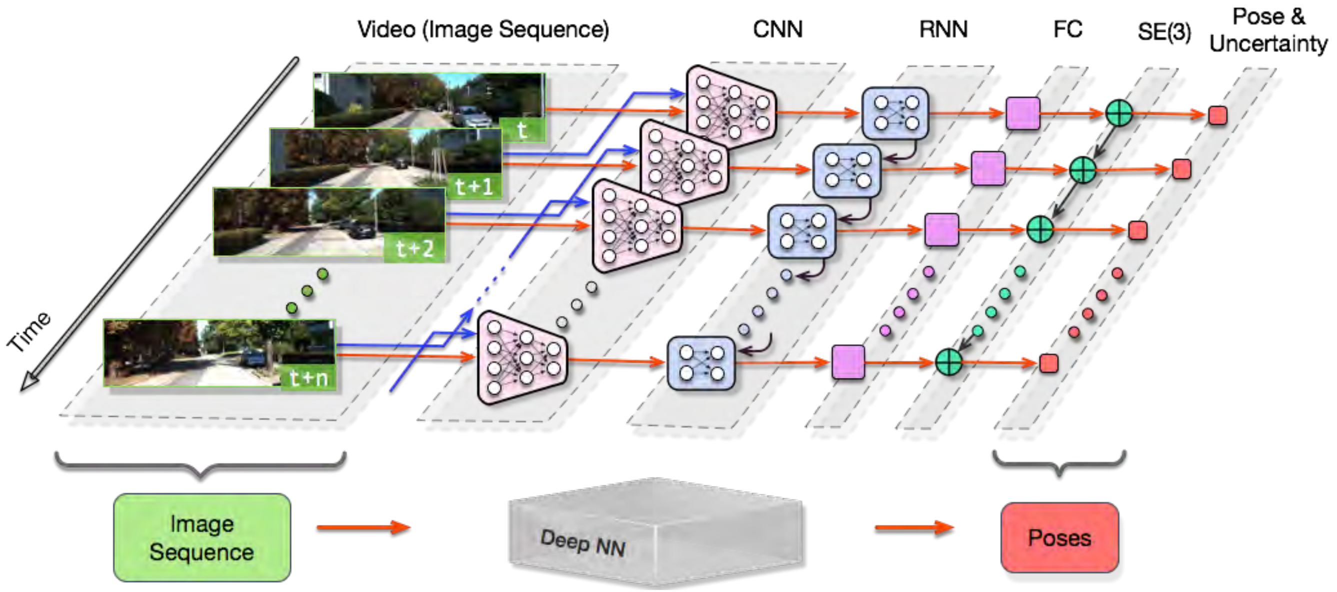
# Deep Learning

- curse of data
  - high-dimensional: raw images
  - big: sensor data from hundreds of thousands of robots/devices



# Architecture

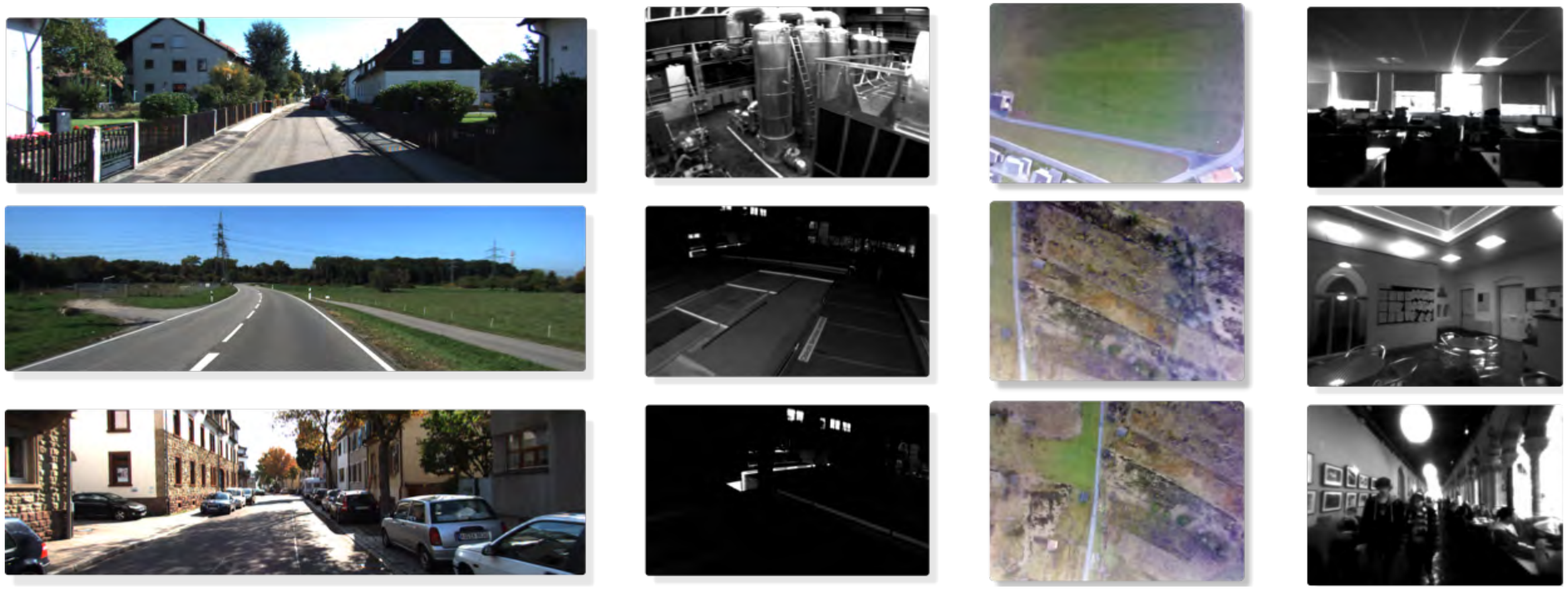
- achieve monocular Visual Odometry (VO) in an end-to-end, sequence-to-sequence manner based on Deep Learning, i.e., directly estimating poses from a sequence of raw RGB images
- leverage a large number of images
- feature extraction + sequential learning




**[Patent]** “Visual odometry through deep neural networks.”
**[IJRR'17]** “End-to-end, sequence-to-sequence probabilistic visual odometry through deep neural networks.”
**[ICRA'17]** “DeepVO: Towards end-to-end visual odometry with deep recurrent convolutional neural networks.”

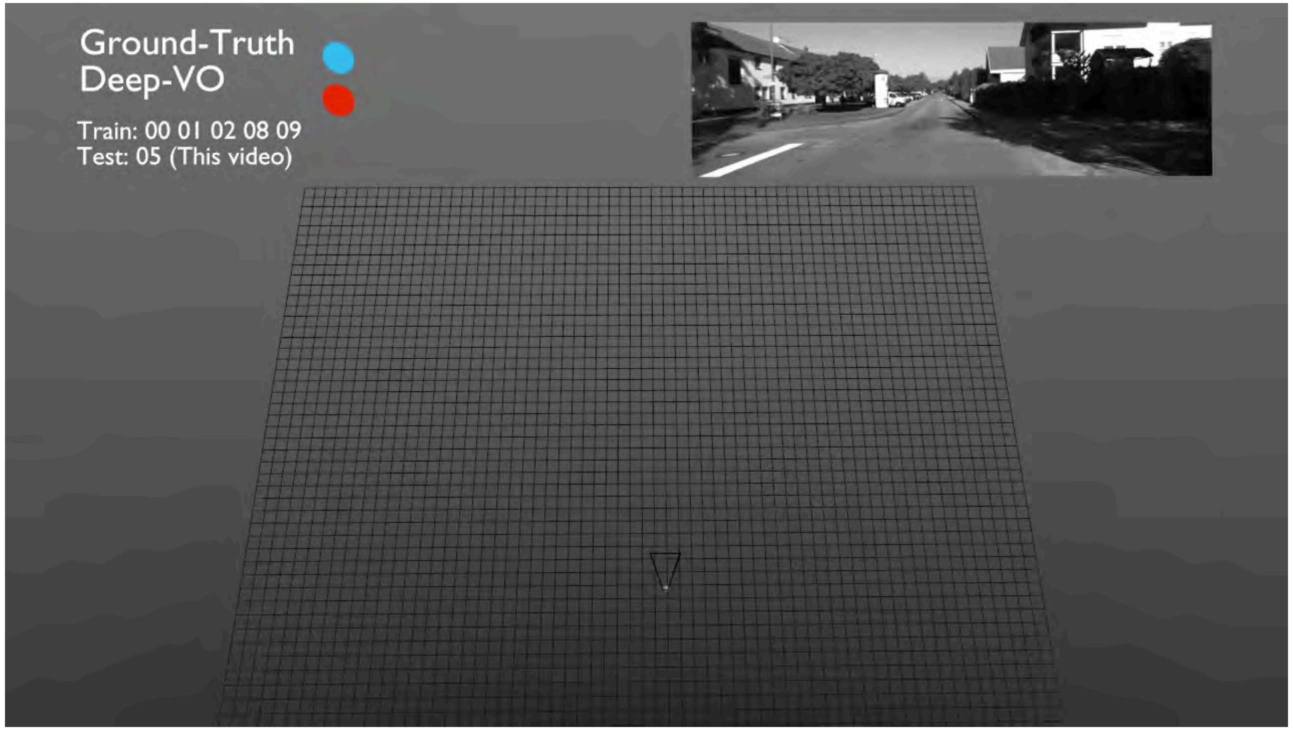
# Experiments

- Testing scenarios: outdoor driving, indoor Micro Aerial Vehicle, indoor pedestrian, outdoor fixed-wing UAV
- Comparison (VO version):
  - VISO2: sparse feature based VO
  - ORB-SLAM: state-of-the-art sparse feature based visual SLAM
  - LSD-SLAM: state-of-the-art direct method based visual SLAM



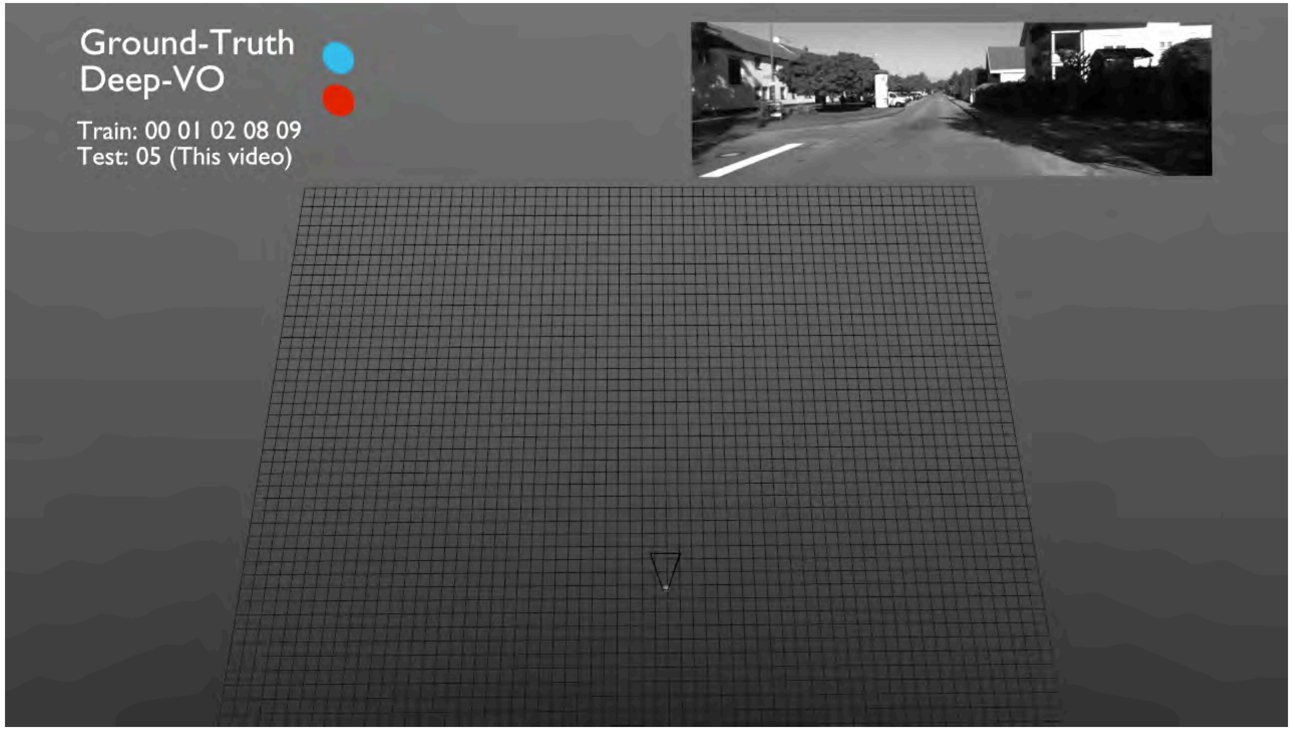
# Outdoor Driving

- KITTI VO/SLAM Benchmark and Raw Dataset
- Trained on
  - 5 of VO training sequences: test on other training sequences
  - all training sequences: test on testing and raw sequences

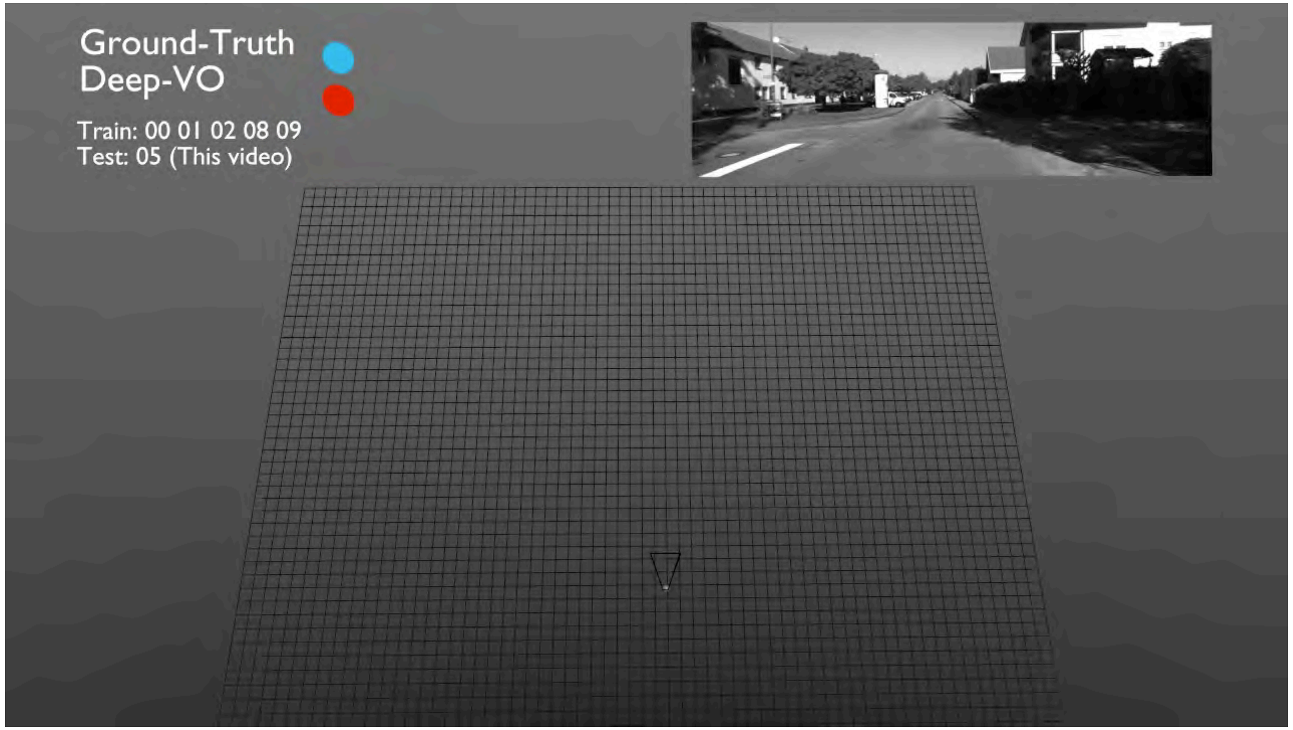
# Outdoor Driving

- KITTI VO/SLAM Benchmark and Raw Dataset
- Trained on
  - 5 of VO training sequences: test on other training sequences
  - all training sequences: test on testing and raw sequences

# Outdoor Driving

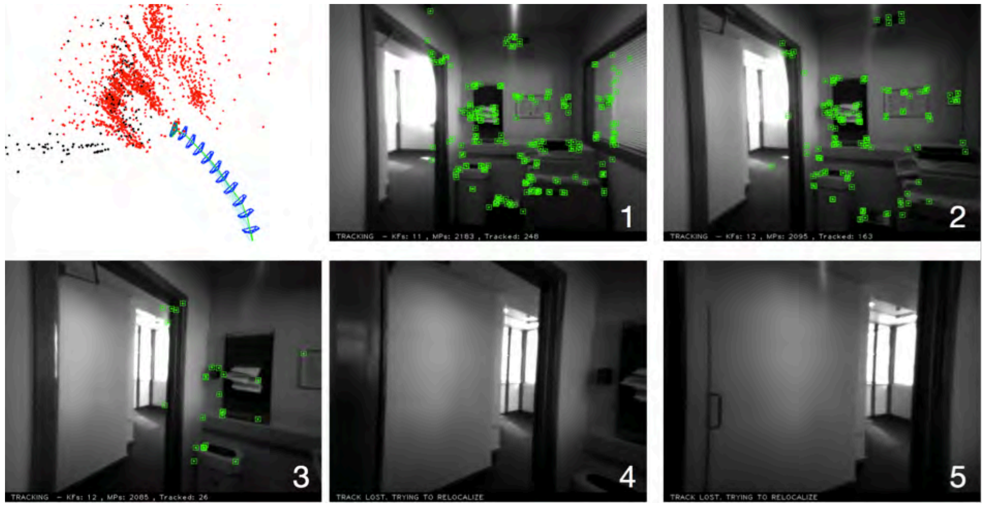
- KITTI VO/SLAM Benchmark and Raw Dataset
- Trained on
  - 5 of VO training sequences: test on other training sequences
  - all training sequences: test on testing and raw sequences

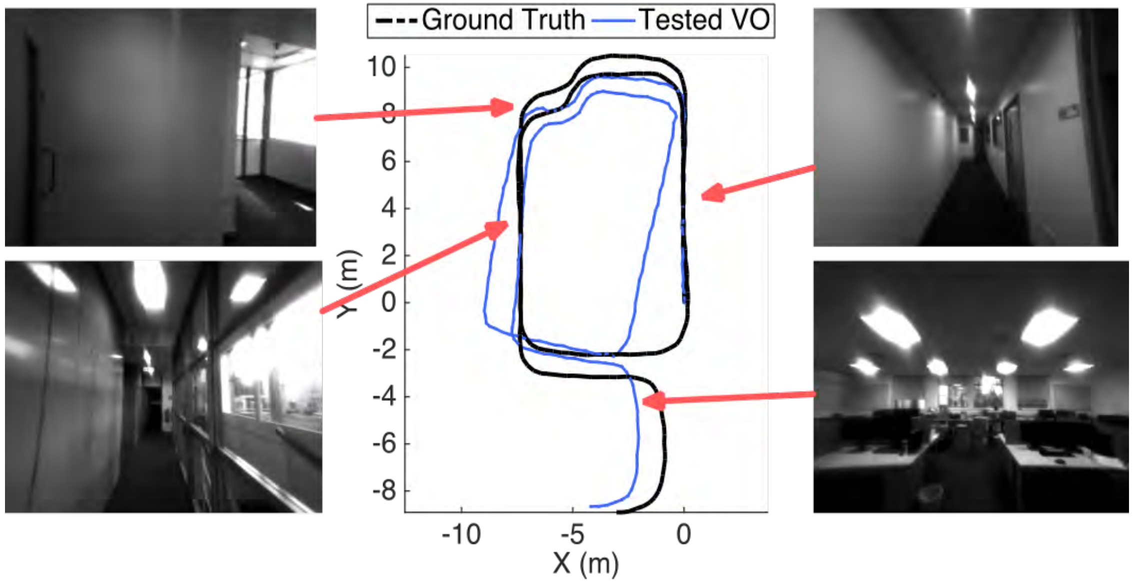
# Indoor Motion: Test in Office Building

- Geometry based methods
  - prone to lose tracking
  - texture-less: corridor, white wall
  - agile motion: fast turn

ORB-SLAM: lose tracking

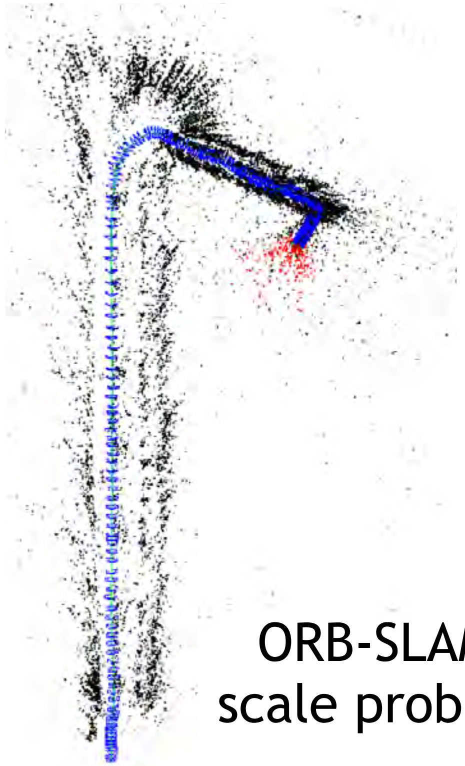



Deep-VO

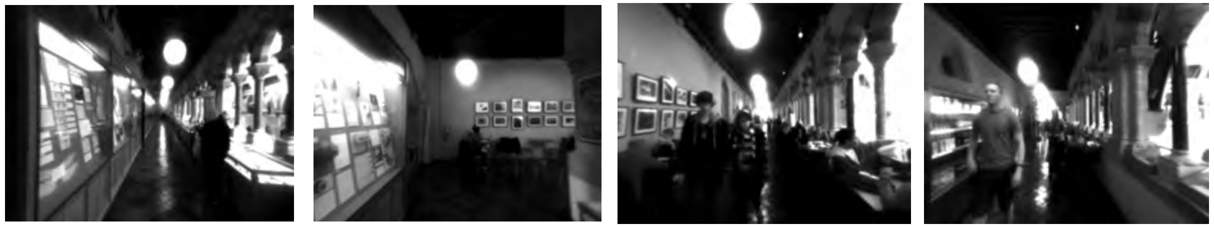


# Indoor Motion: Test in Museum

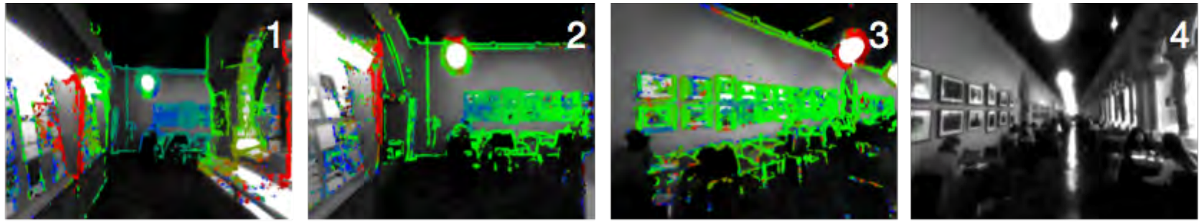
- Natural History Museum
    - Challenging lighting conditions
  - Café with walking people
- Deep-VO: see motion in video



ORB-SLAM:
scale problem

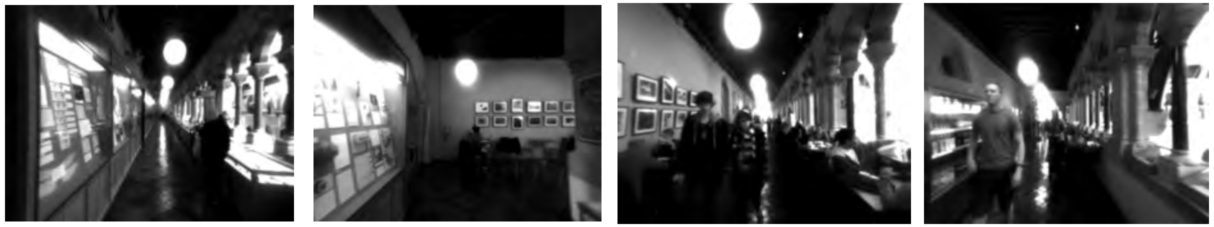


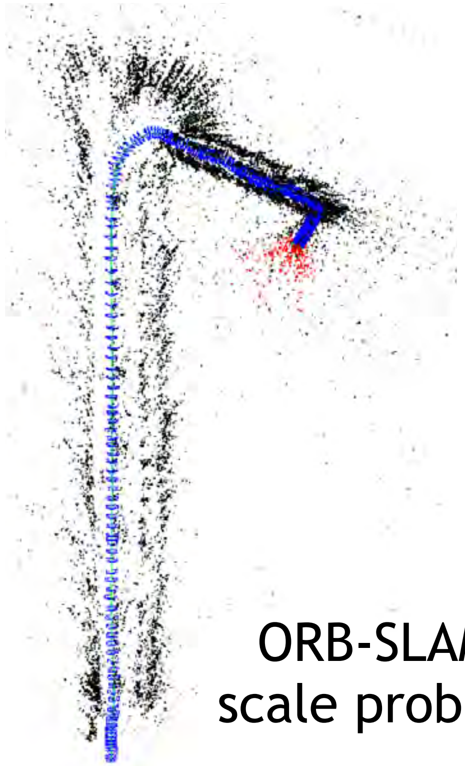
LSD-SLAM: lose tracking



# Indoor Motion: Test in Museum

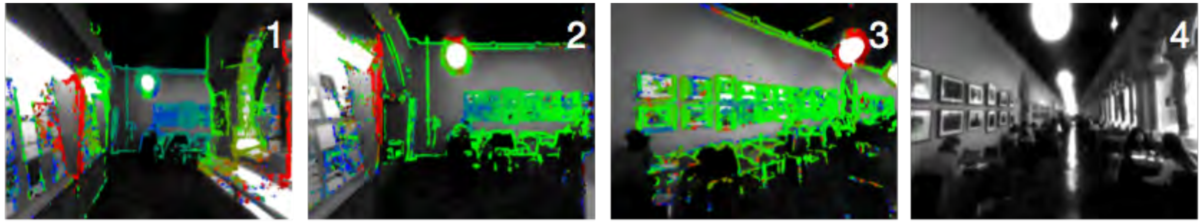
- Natural History Museum
  - Challenging lighting conditions
  - Café with walking people
- Deep-VO: see motion in video





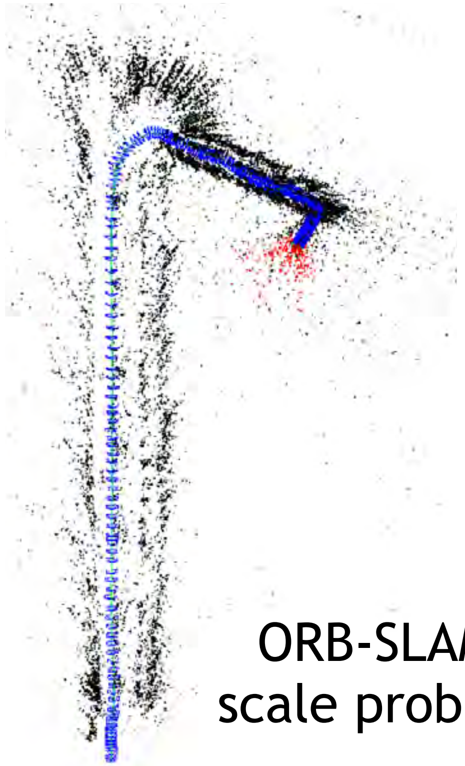
ORB-SLAM: scale problem

LSD-SLAM: lose tracking

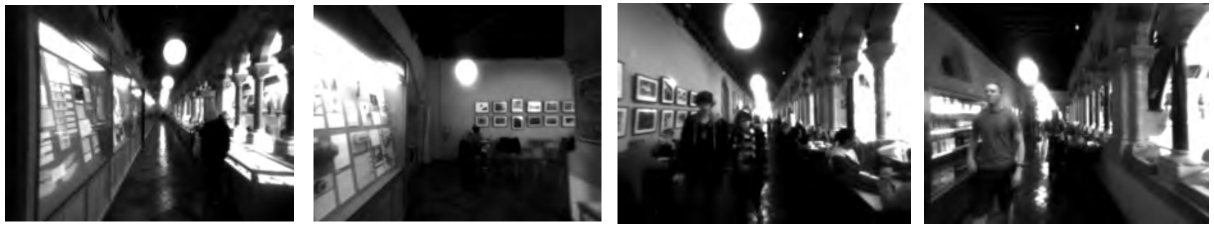


# Indoor Motion: Test in Museum

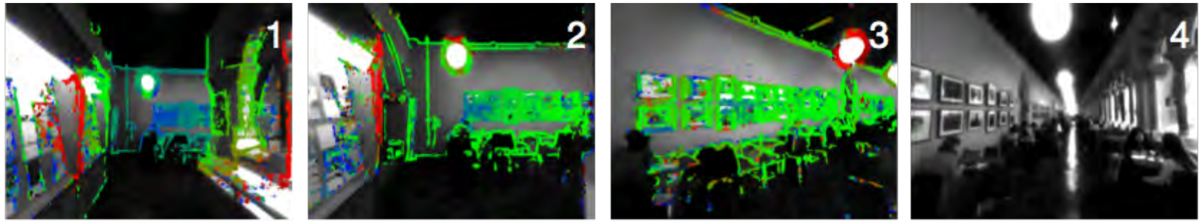
- Natural History Museum
    - Challenging lighting conditions
  - Café with walking people
- Deep-VO: see motion in video



ORB-SLAM:
scale problem



LSD-SLAM: lose tracking



# Unsupervised Learning

supervised learning:
annotating data is expensive/ labour-intensive!

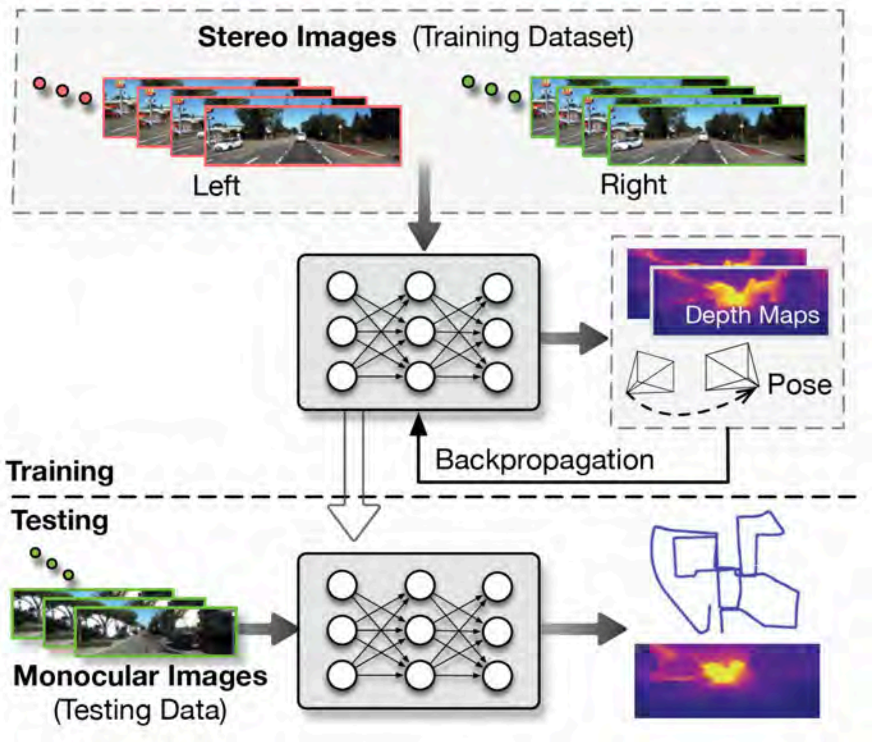
# Unsupervised Learning

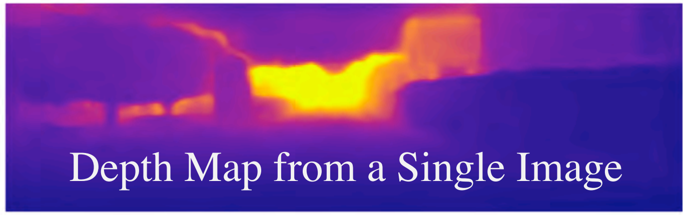
supervised learning:
annotating data is expensive/ labour-intensive!

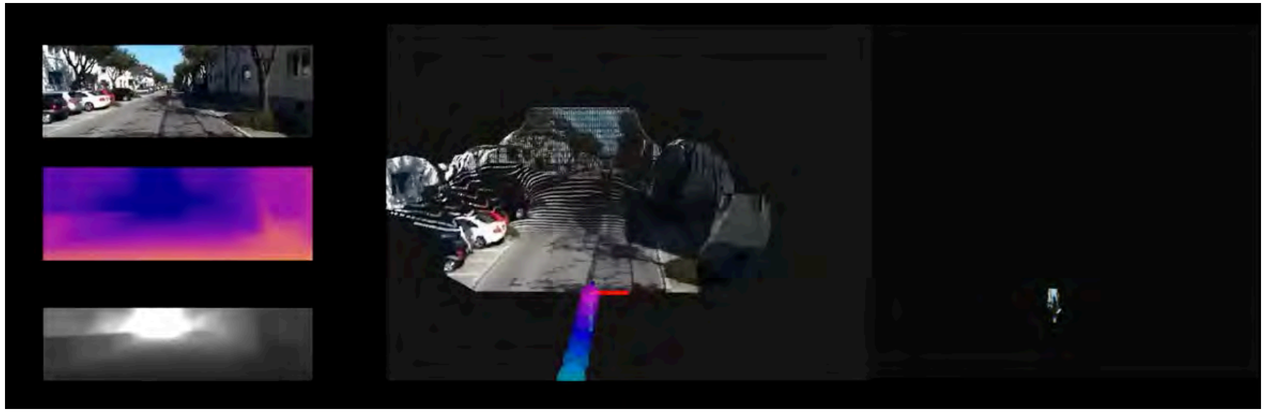
unsupervised learning?
lifelong leaning?

# Visual Odometry with Unsupervised Learning

- unsupervised learning for VO through visual geometry
- Training: only a stereo camera
- Testing: a monocular camera



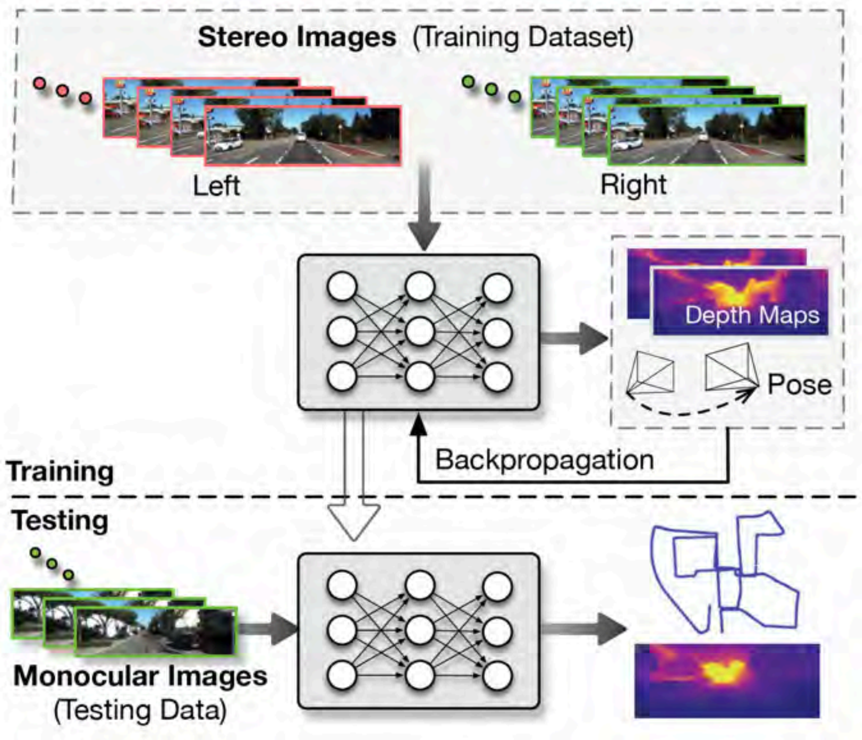



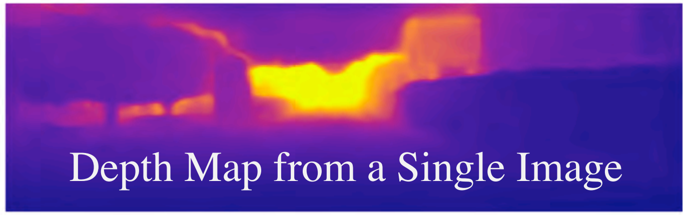



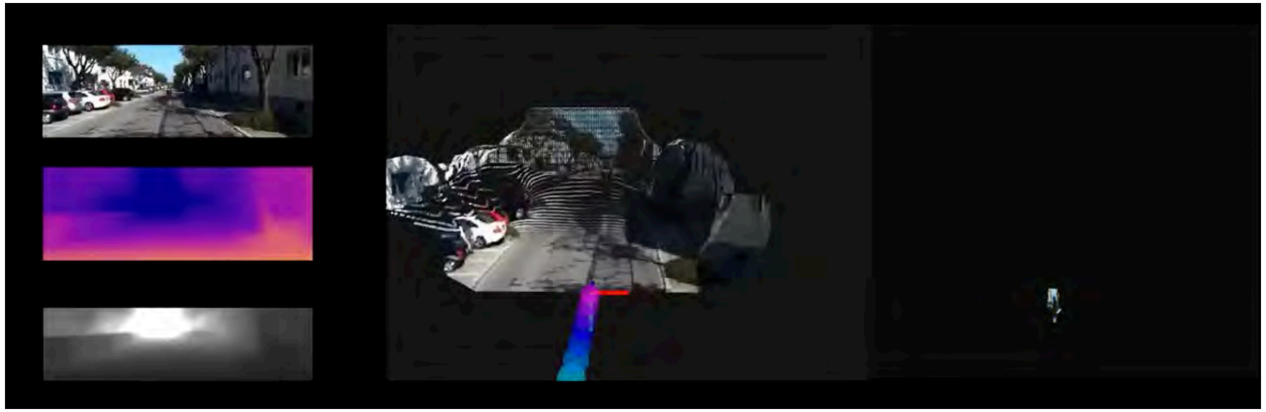
**[ICRA'18]** "UnDeepVO: Monocular Visual Odometry through Unsupervised Deep Learning."

# Visual Odometry with Unsupervised Learning

- unsupervised learning for VO through visual geometry
- Training: only a stereo camera
- Testing: a monocular camera



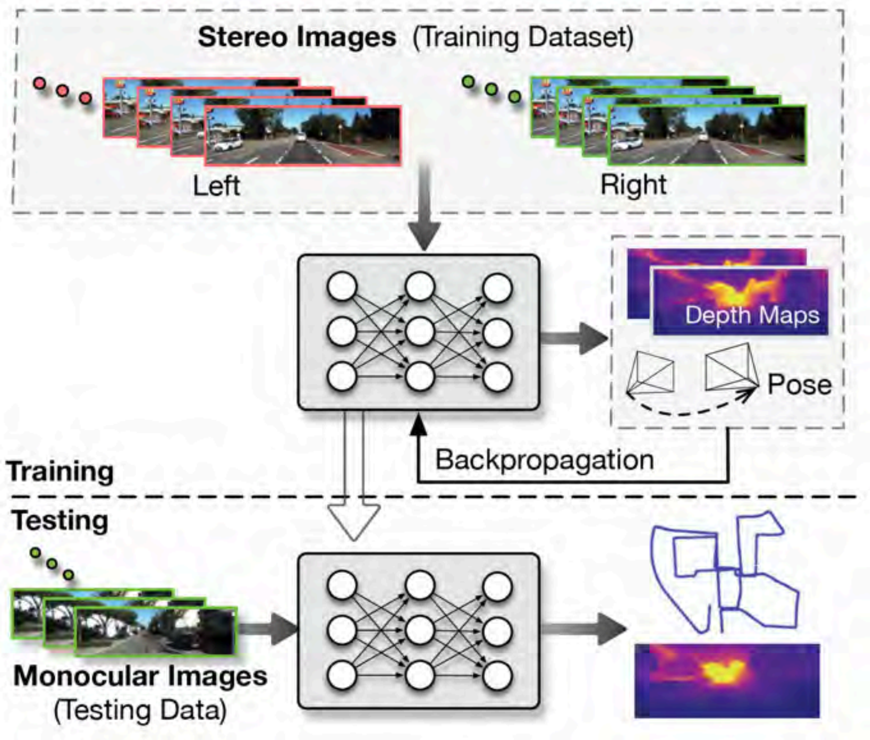



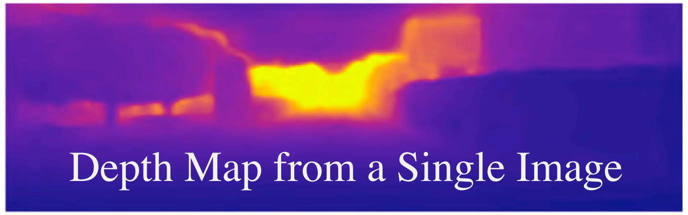



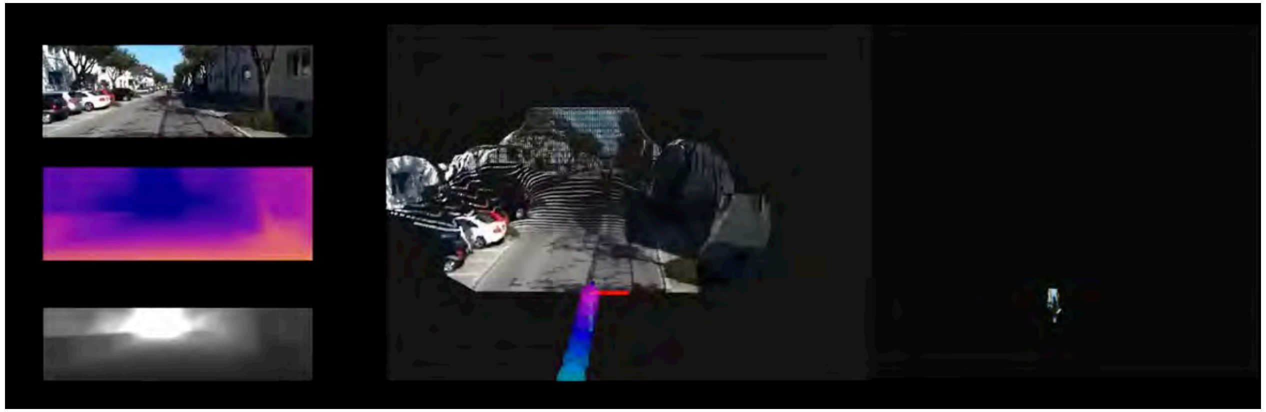
**[ICRA'18]** "UnDeepVO: Monocular Visual Odometry through Unsupervised Deep Learning."

# Visual Odometry with Unsupervised Learning

- unsupervised learning for VO through visual geometry
- Training: only a stereo camera
- Testing: a monocular camera



**[ICRA'18]** "UnDeepVO: Monocular Visual Odometry through Unsupervised Deep Learning."

# Better Accuracy and Robustness

- better camera
  - wide dynamic range, global shutter, high frame rate, wide-angle, etc.

# Better Accuracy and Robustness

- better camera
  - wide dynamic range, global shutter, high frame rate, wide-angle, etc.
- better algorithm
  - outliers: random sample consensus (RANSAC)

# Better Accuracy and Robustness

- better camera
  - wide dynamic range, global shutter, high frame rate, wide-angle, etc.
- better algorithm
  - outliers: random sample consensus (RANSAC)
- drift problem
  - loop closure detection -> SLAM
  - graph optimisation

# Better Accuracy and Robustness

- better camera
  - wide dynamic range, global shutter, high frame rate, wide-angle, etc.
- better algorithm
  - outliers: random sample consensus (RANSAC)
- drift problem
  - loop closure detection -> SLAM
  - graph optimisation
- scale problem for monocular vision
  - stereo VO or RGB-D
  - other sensors (IMUs), information (fixed height)

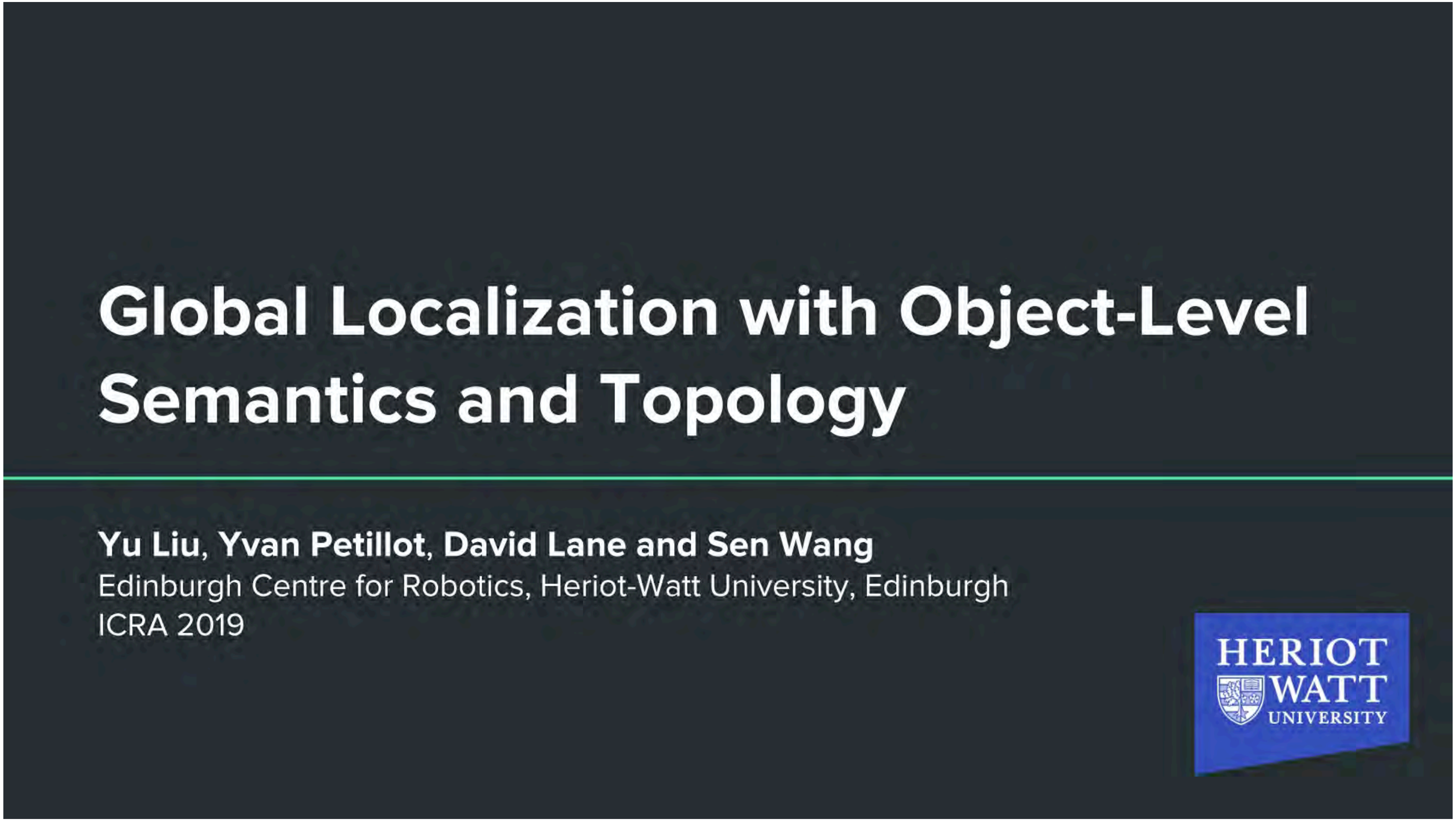
# Future – Semantic/Object Level VO/SLAM

## Global Localization with Object-Level Semantics and Topology

**Yu Liu, Yvan Petillot, David Lane and Sen Wang**
Edinburgh Centre for Robotics, Heriot-Watt University, Edinburgh
ICRA 2019

HERIOT WATT UNIVERSITY

# Future – Semantic/Object Level VO/SLAM



# Future – Semantic/Object Level VO/SLAM

## Global Localization with Object-Level Semantics and Topology

**Yu Liu, Yvan Petillot, David Lane and Sen Wang**
Edinburgh Centre for Robotics, Heriot-Watt University, Edinburgh
ICRA 2019

HERIOT WATT UNIVERSITY