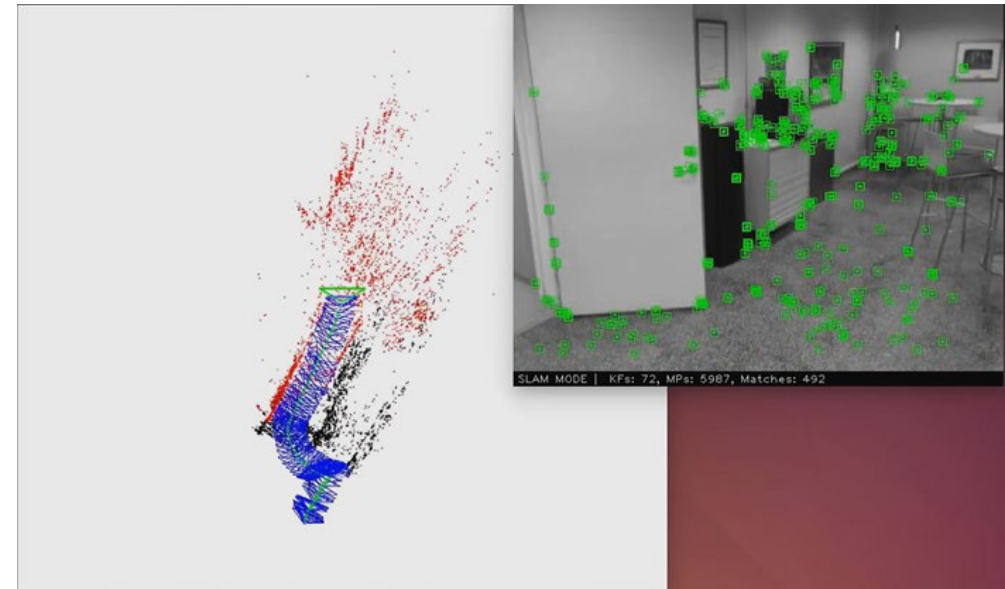


Lecture 13

Visual SLAM and computer vision applications

Trym Vegard Haavardsholm



Today

1. What is Visual SLAM?
2. Short-term, mid-term and long-term tracking
3. Mapping and sensor fusion with factor graphs
4. VSLAM backend strategies
5. VSLAM systems
6. Example application

Part I

WHAT IS VISUAL SLAM?

What is SLAM?

What is SLAM?

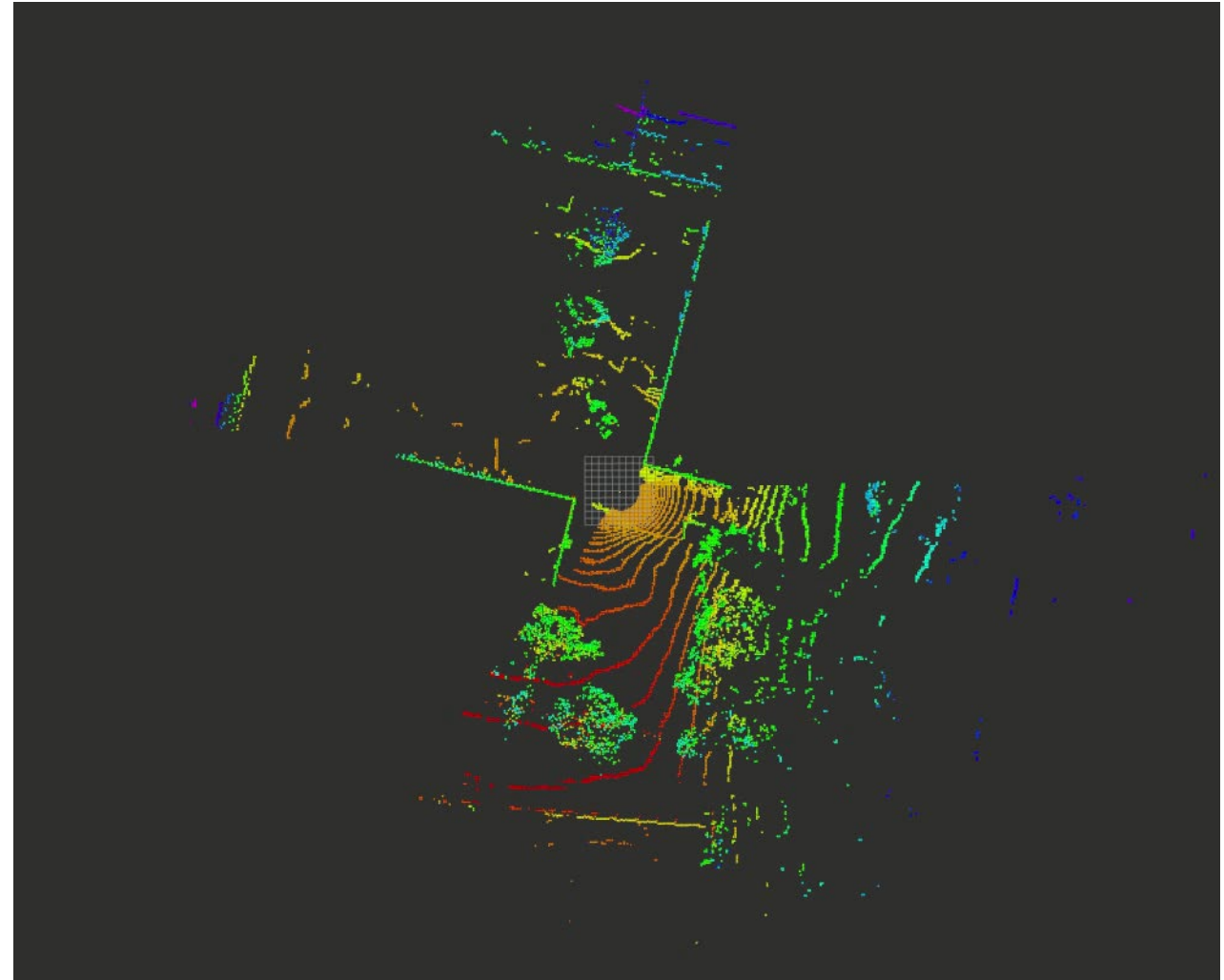
Simultaneous localisation and mapping

What is SLAM?

Simultaneous localisation and mapping

Simultaneous

- estimation of the state of a robot using on-board sensors
- construction of a map of the environment that the sensors are perceiving

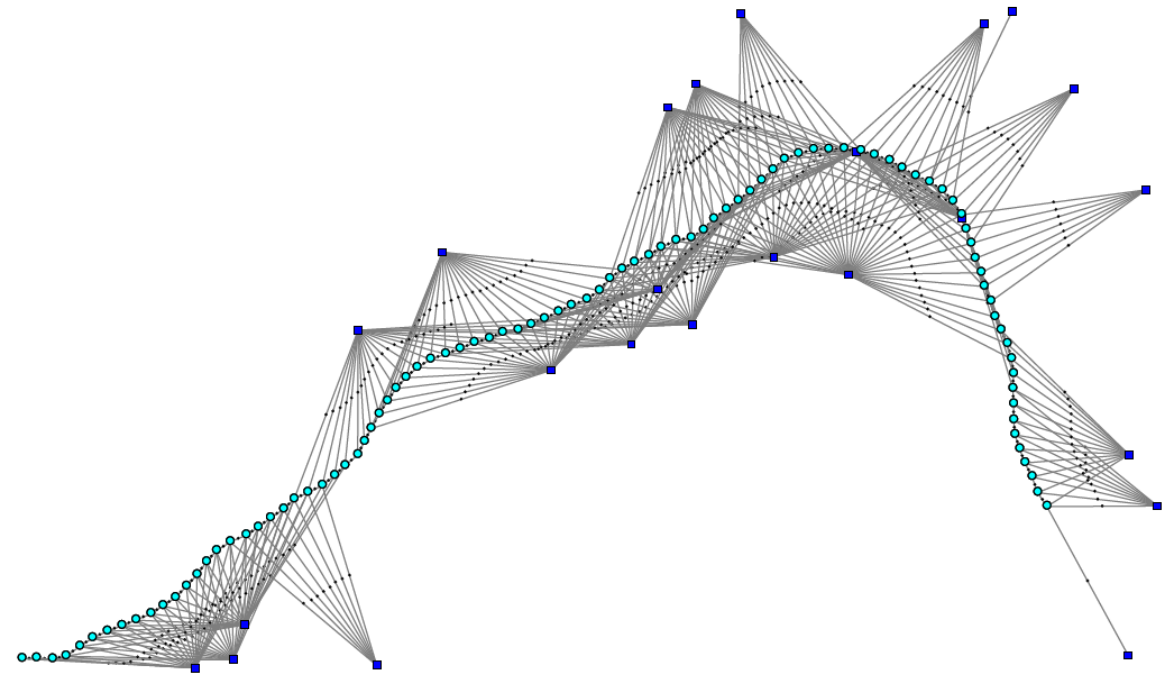


What is SLAM?

Simultaneous localisation and mapping

Simultaneous

- **mapping:**
Continuously expanding and optimising a consistent map while exploring the environment
- **localisation:**
Localisation within the map



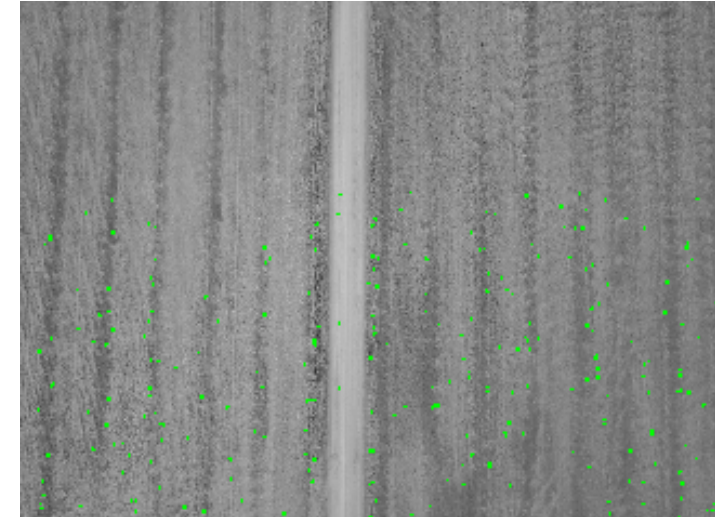
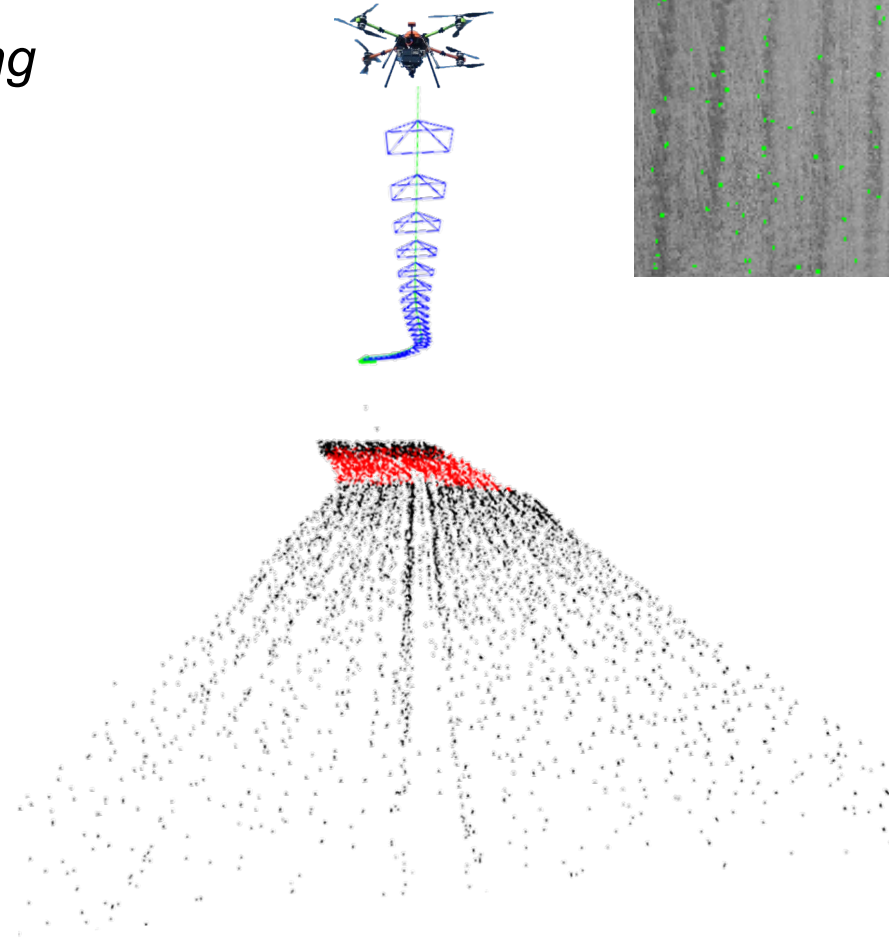
Jing Dong "[GTSAM 4.0 Tutorial](#)" License CC BY-NC-SA 3.0

What is Visual SLAM?

Visual simultaneous localisation and mapping

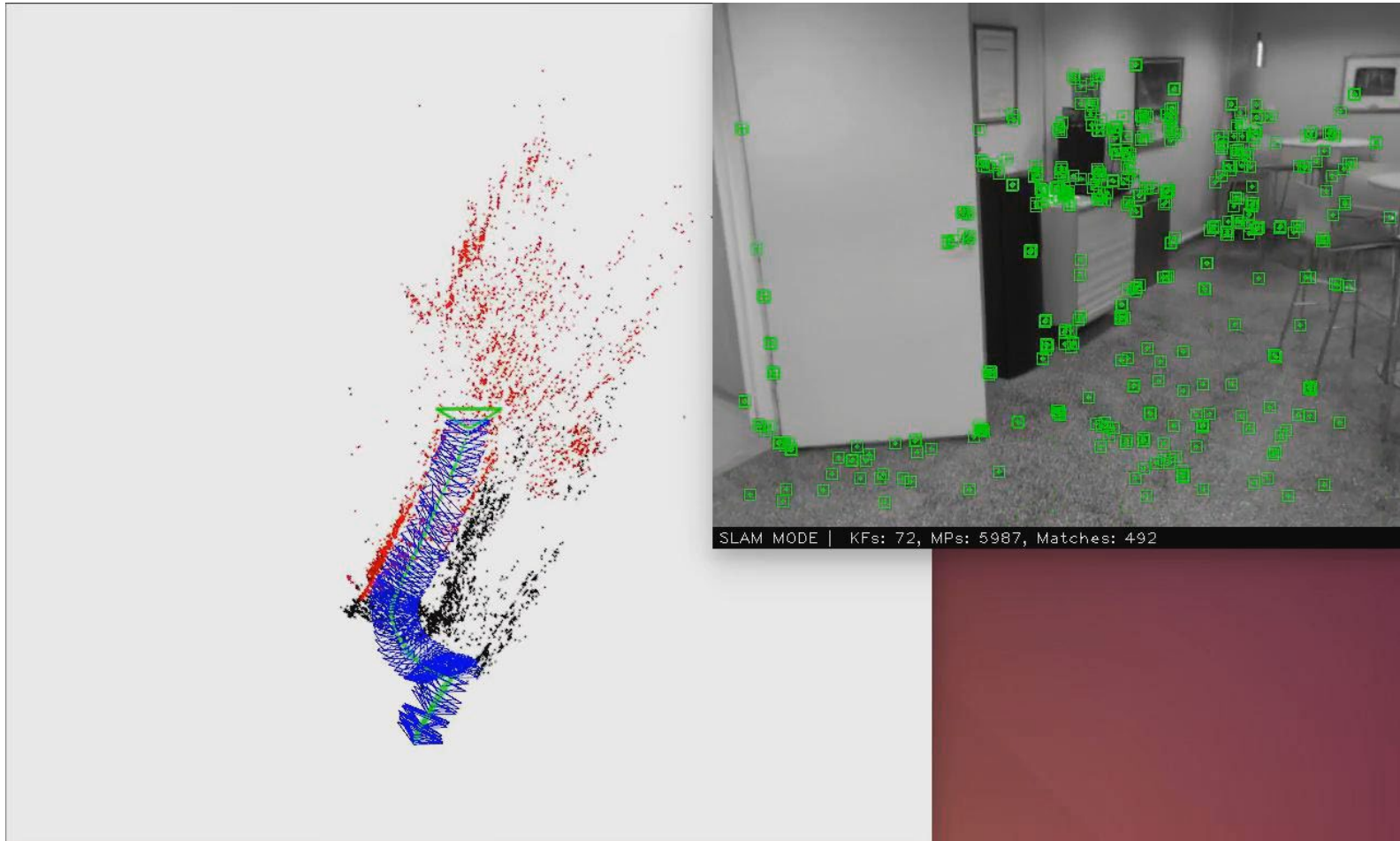
Simultaneous

- **mapping:**
Continuously expanding and optimising a consistent map while exploring the environment
- **localisation (tracking):**
Localisation within the map
(tracking the map in image frames)



How can we do visual SLAM?

Visual SLAM example

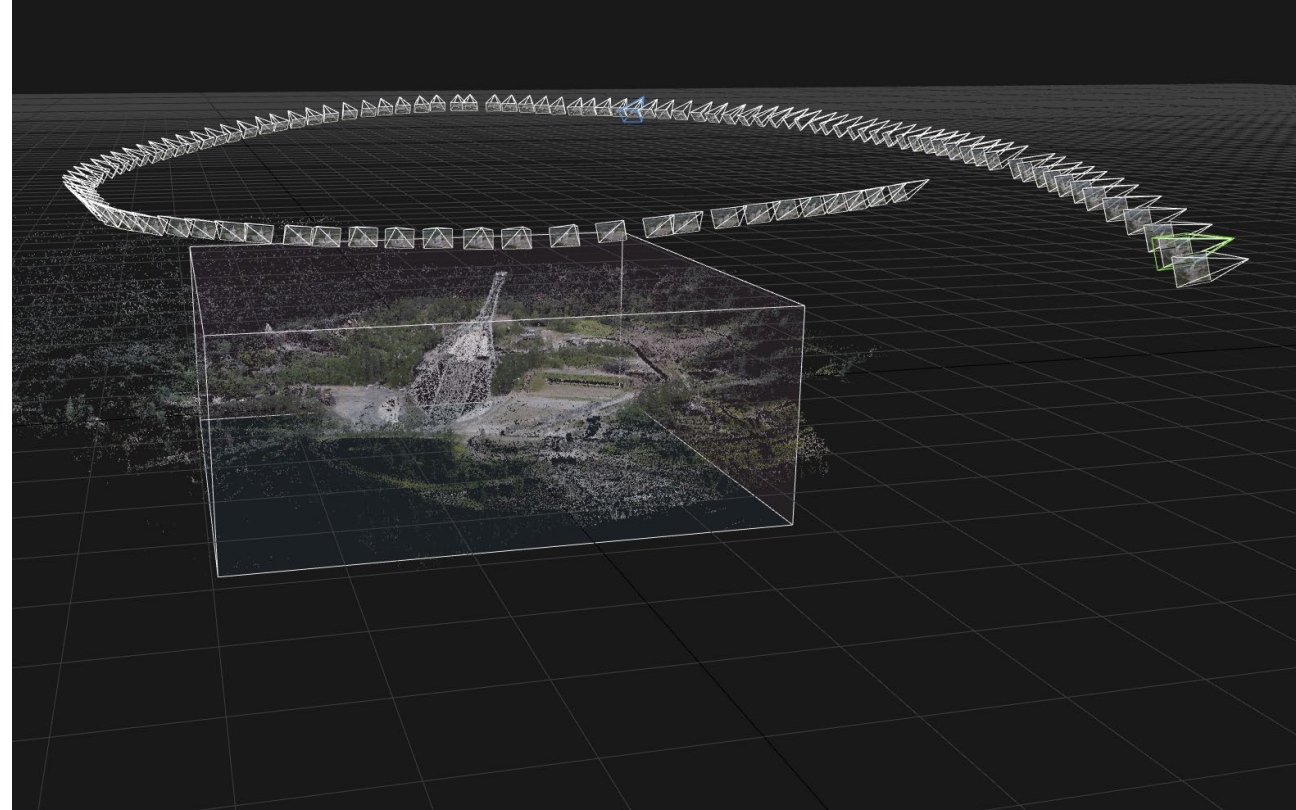


What is the map?

What is the map?

A model of the environment that lets us

- limit the localisation error by recognising previously visited areas
- (support other tasks, such as obstacle avoidance and path planning)

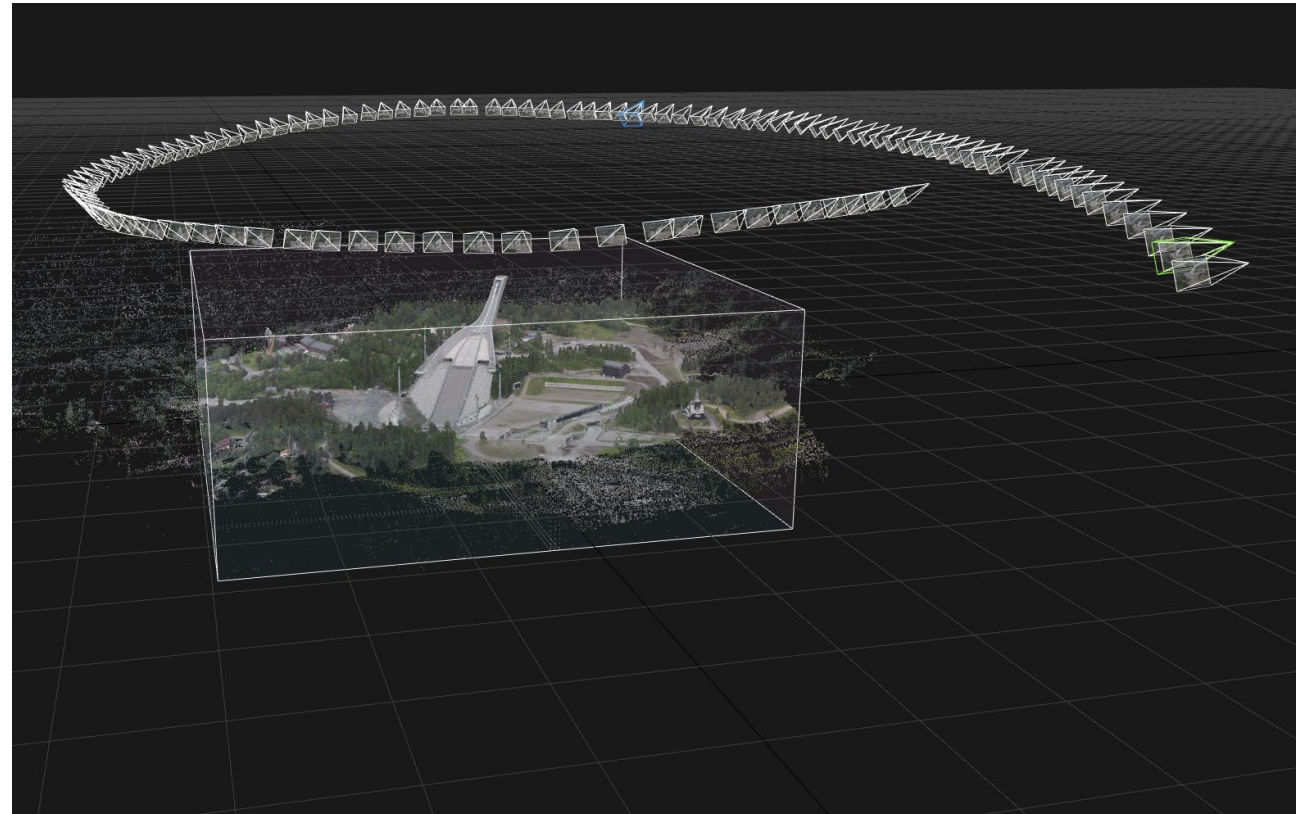


What is the map?

A model of the environment that lets us

- limit the localisation error by recognising previously visited areas
- (support other tasks, such as obstacle avoidance and path planning)

Maybe best left as auxiliary processing?



Examples of map representations

Feature-based metric maps

Representation based on:

- Graphs over *keyframes*
- Kd-trees over *points*

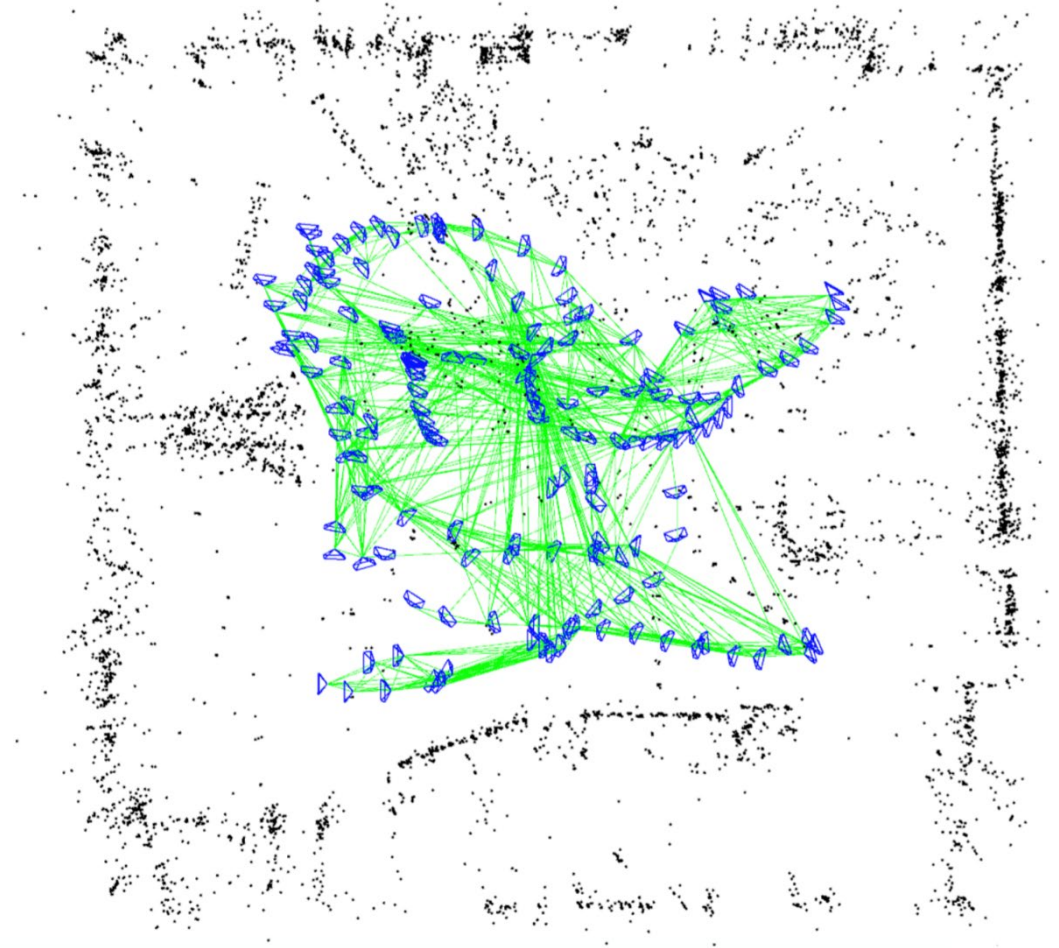


Image: Cadena, C., et al. (2016). Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age. *IEEE Transactions on Robotics*, 32(6), 1309–1332

Mur-Artal, R., Montiel, J. M. M., & Tardos, J. D. (2015). ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*, 31(5), 1147–1163. <https://doi.org/10.1109/TRO.2015.2463671>

Examples of map representations

Dense metric maps

[DTAM:](#)
[Dense Tracking and Mapping in Real-Time](#)



Image: Cadena, C., et al. (2016). Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age. *IEEE Transactions on Robotics*, 32(6), 1309–1332

Newcombe, R. A., Lovegrove, S. J., & Davison, A. J. (2011). DTAM: Dense tracking and mapping in real-time. In 2011 International Conference on Computer Vision (pp. 2320–2327). IEEE

Examples of map representations

Dense metric maps

[DTAM:](#)
[Dense Tracking and Mapping in Real-Time](#)

Representation example:



voxblox

<https://voxblox.readthedocs.io/en/latest/>

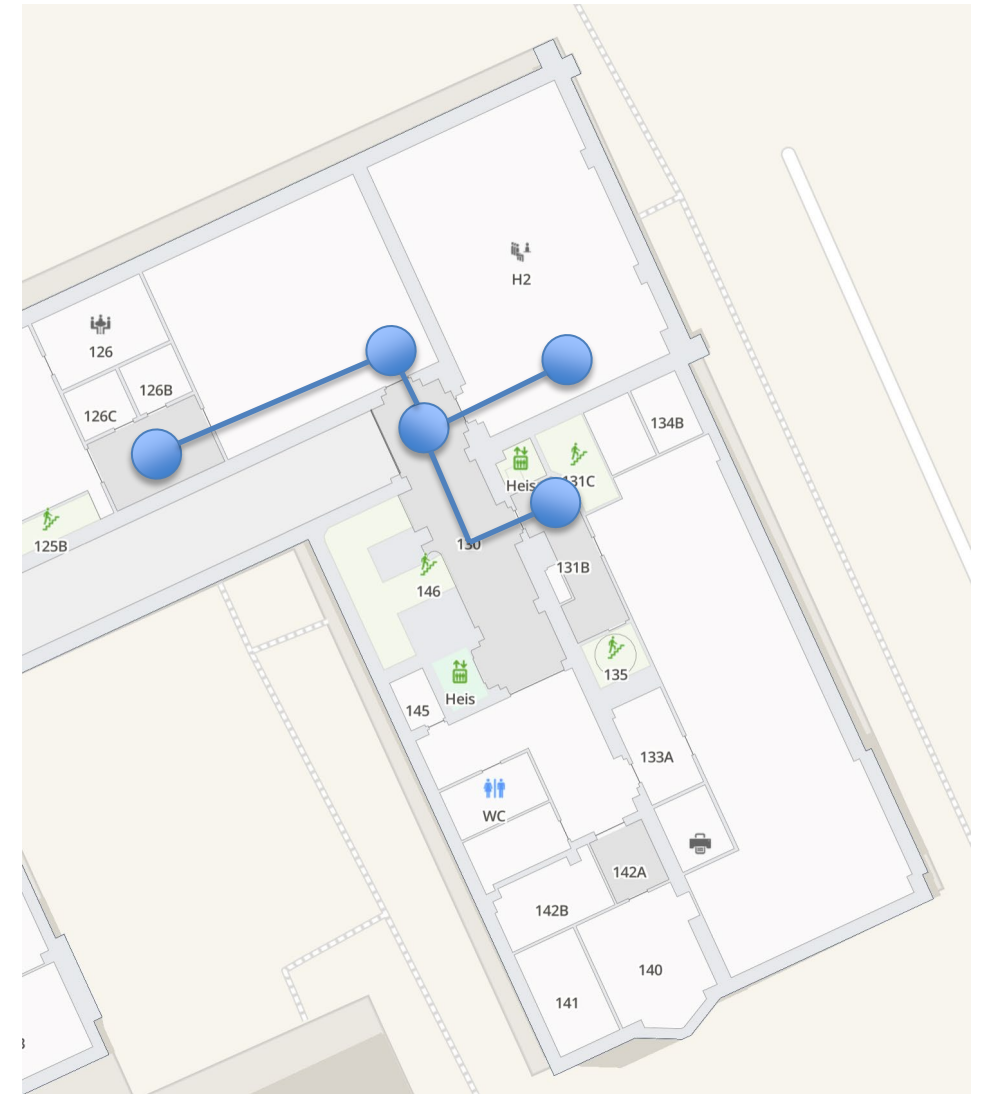


Image: Cadena, C., et al. (2016). Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age. *IEEE Transactions on Robotics*, 32(6), 1309–1332

Newcombe, R. A., Lovegrove, S. J., & Davison, A. J. (2011). DTAM: Dense tracking and mapping in real-time. In 2011 International Conference on Computer Vision (pp. 2320–2327). IEEE

Examples of map representations

Topological maps



Examples of map representations

Topological maps

FABMAP



Image: YouTube: ORI - Oxford Robotics Institute

Cummins, M., & Newman, P. (2008). FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance. The International Journal of Robotics Research, 27(6), 647–665

Examples of map representations

Topological-metric maps



Examples of map representations

Topological-metric maps

[Visual Teach & Repeat](#)

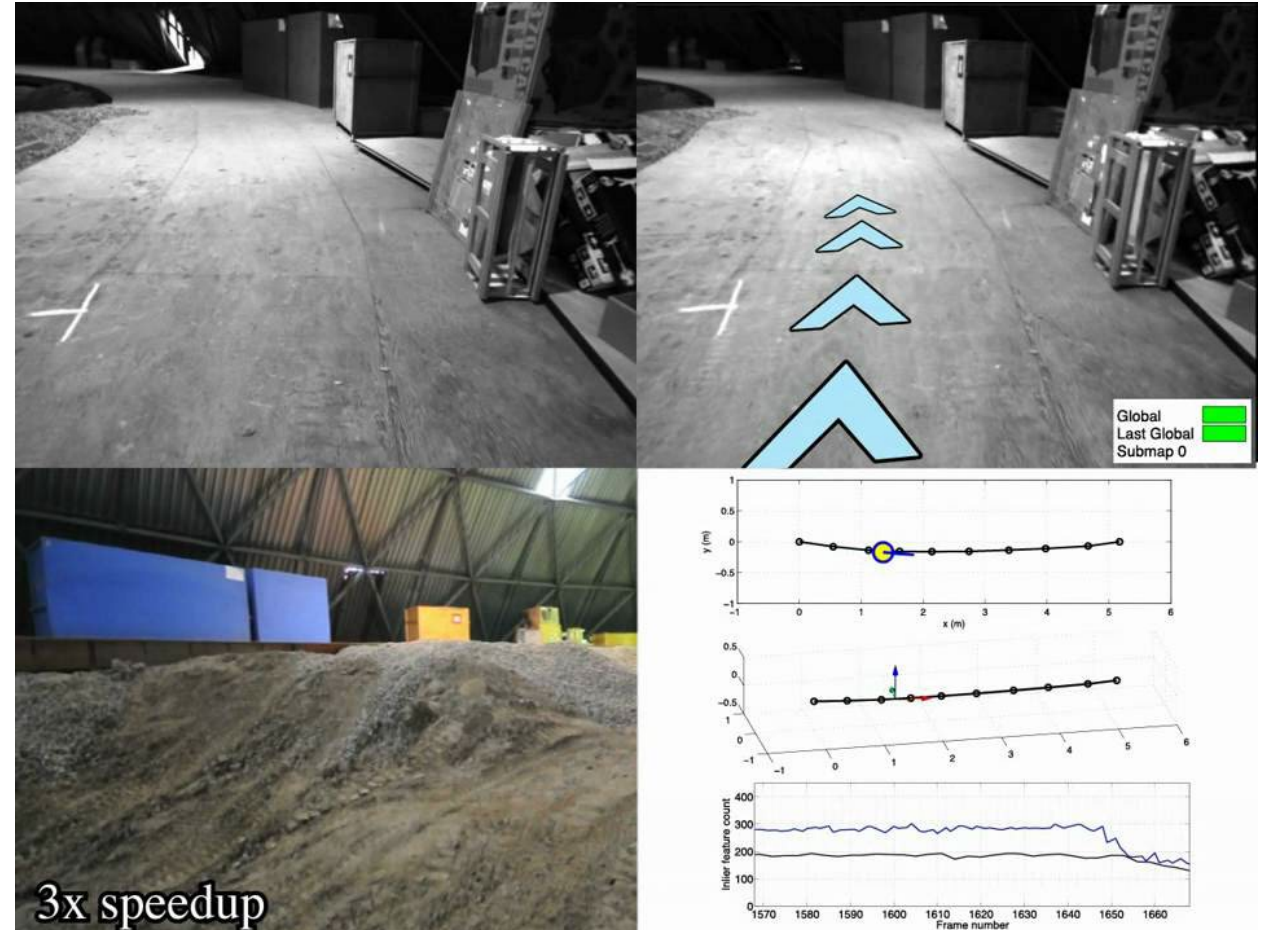
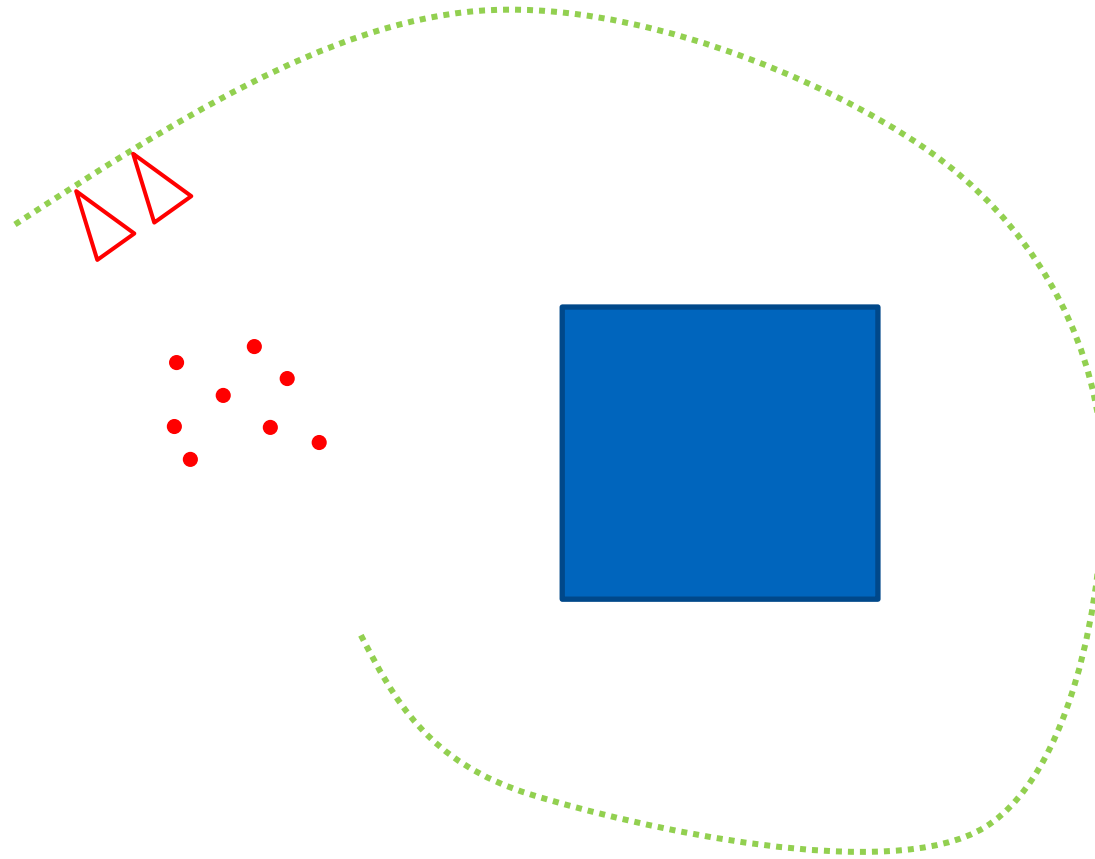


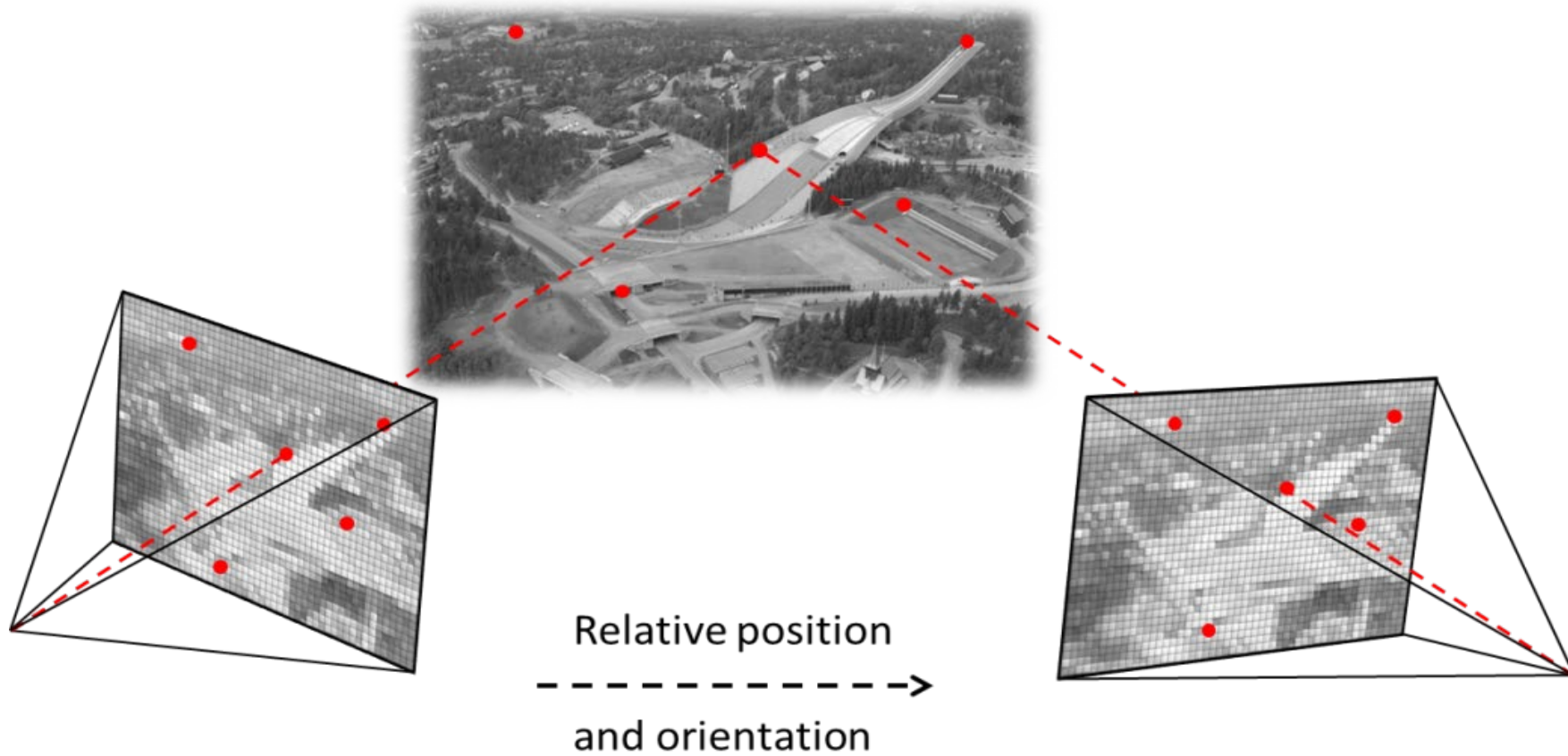
Image: YouTube: utiasASRL

Furgale P T and Barfoot T D. Visual Teach and Repeat for Long-Range Rover Autonomy.
Journal of Field Robotics, special issue on Visual mapping and navigation outdoors, 27(5): 534-560, 2010.

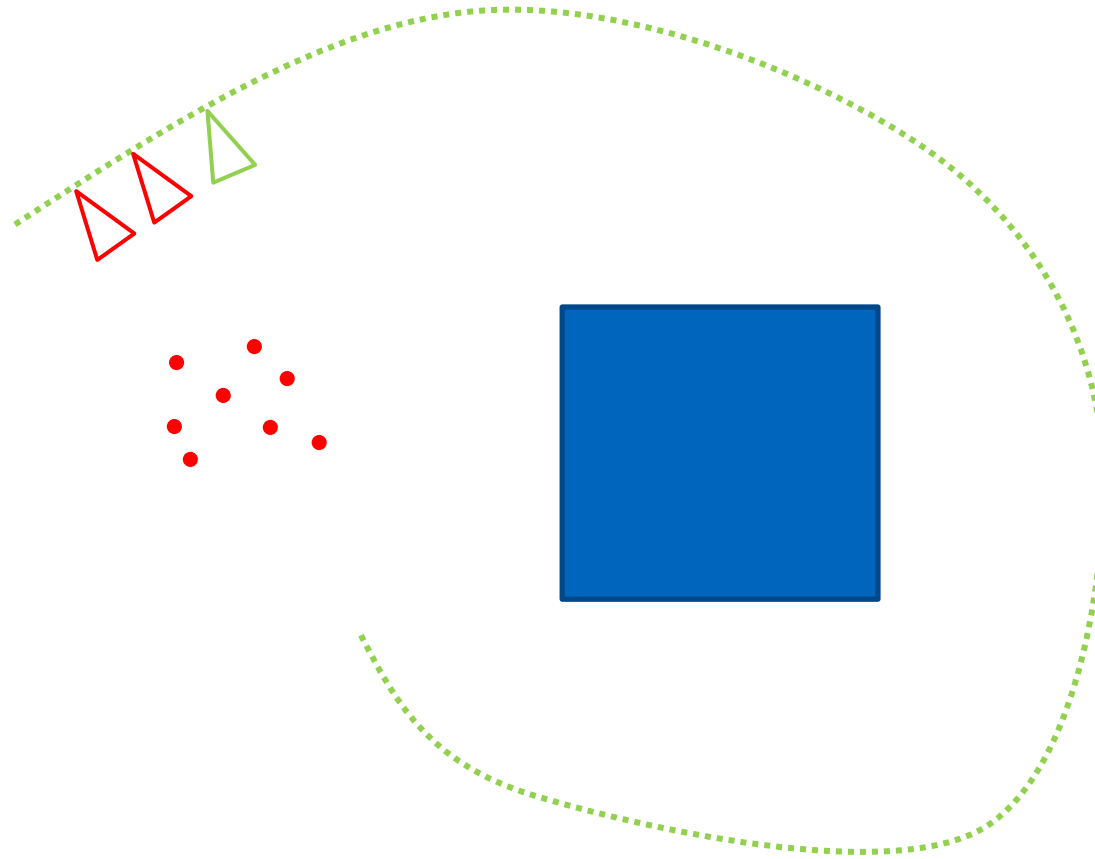
How do we build a map?



Relative pose and 3D from two views



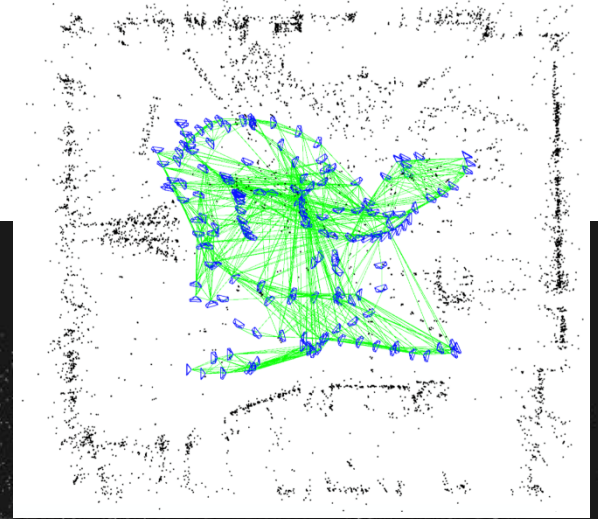
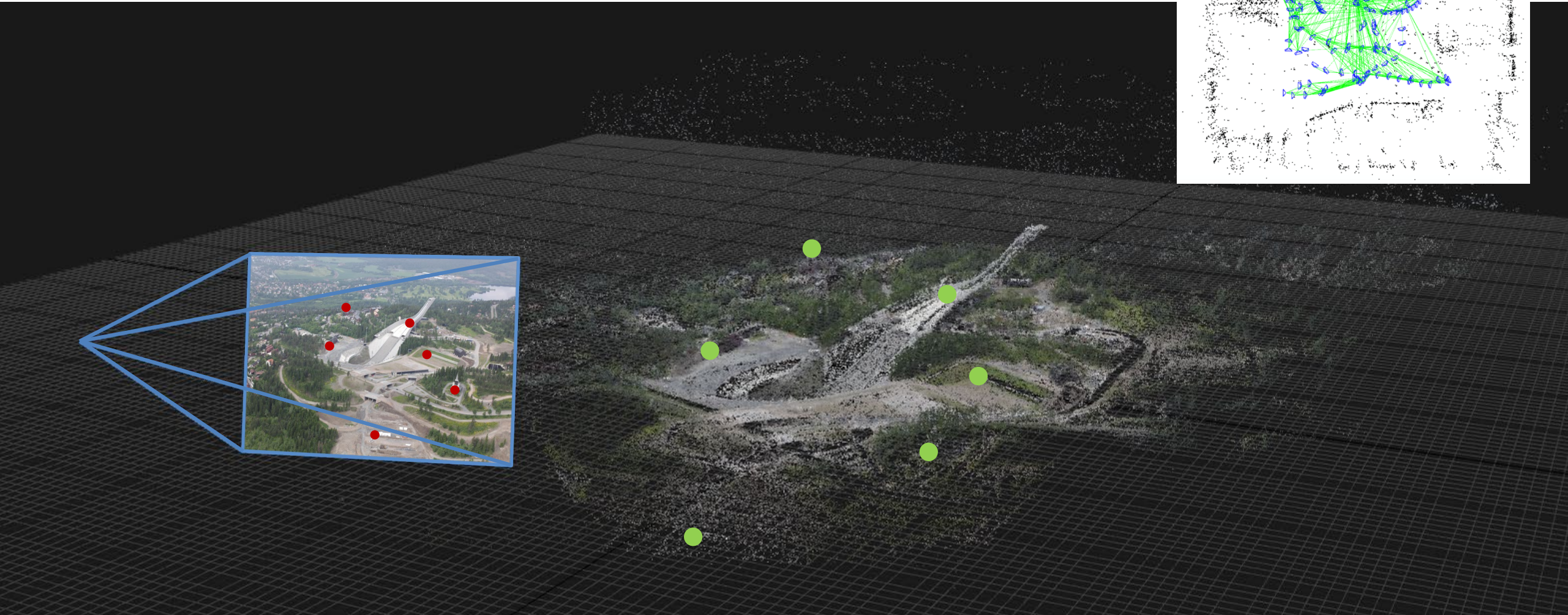
How do we track a map?



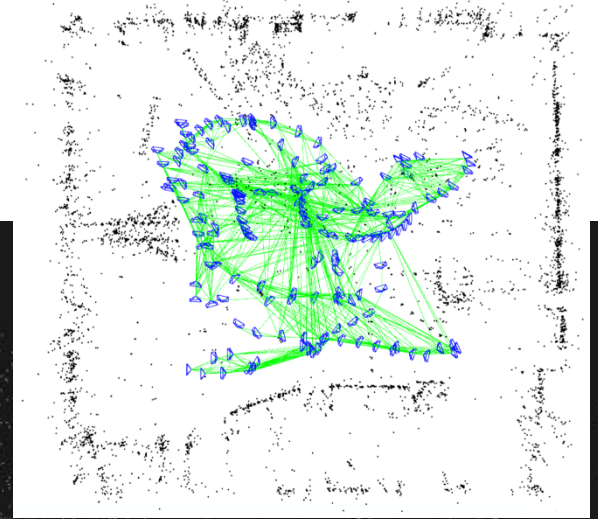
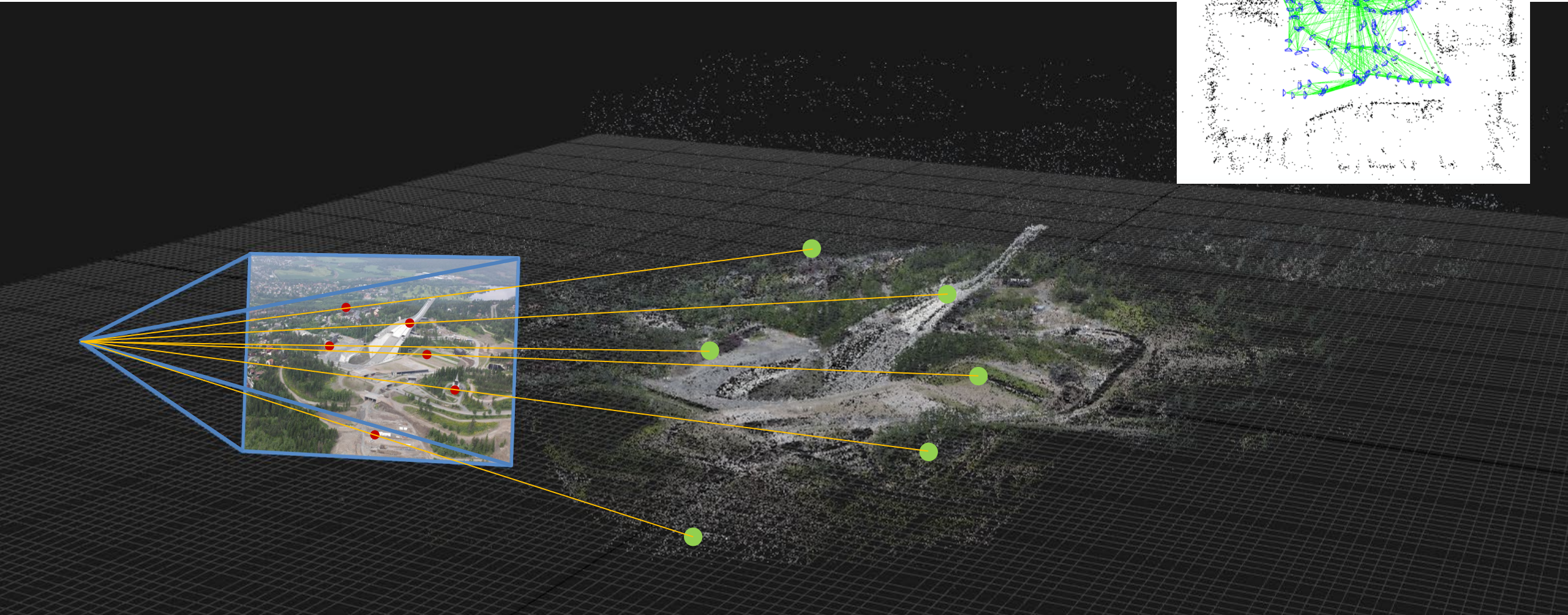
Pose from known 3D map



Pose from point correspondences



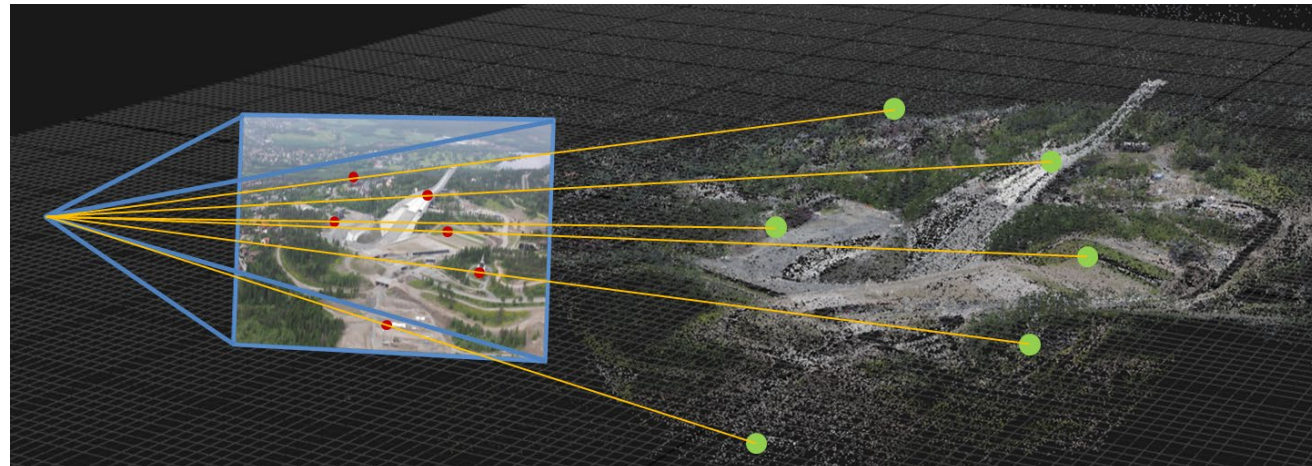
Pose from point correspondences



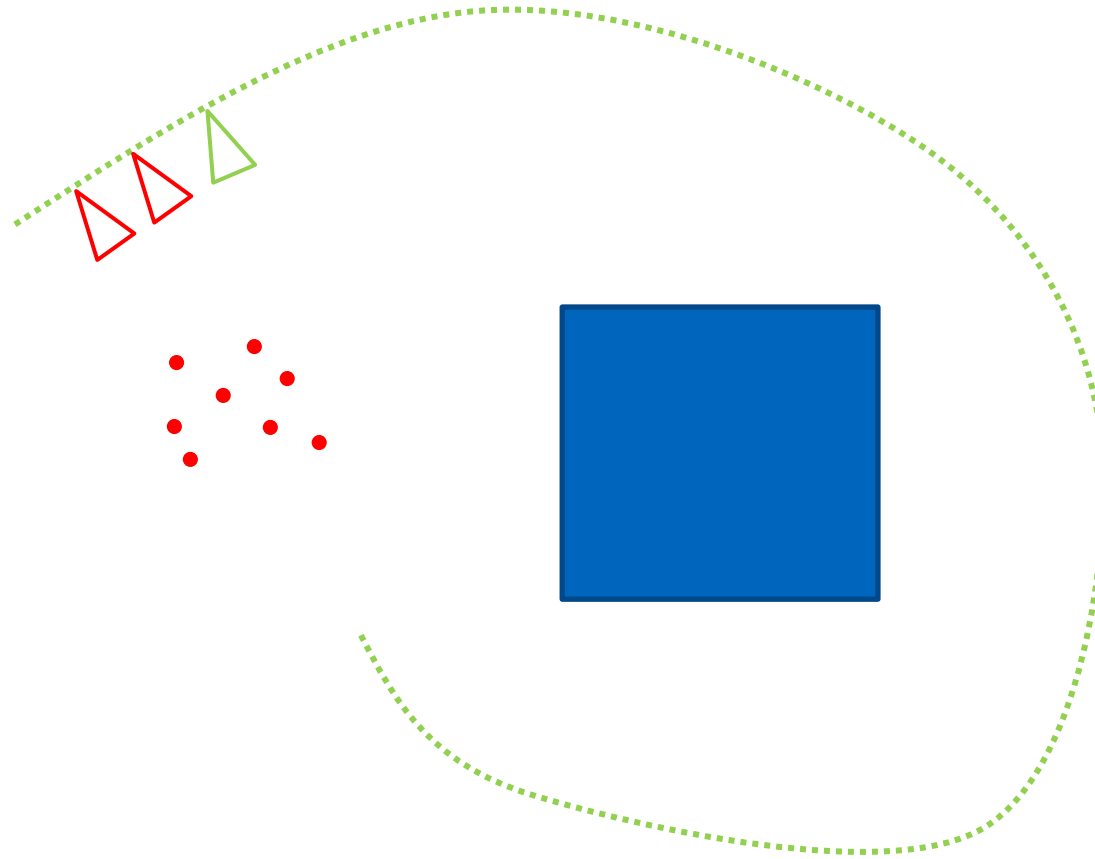
Pose from point correspondences

Minimise *geometric error*

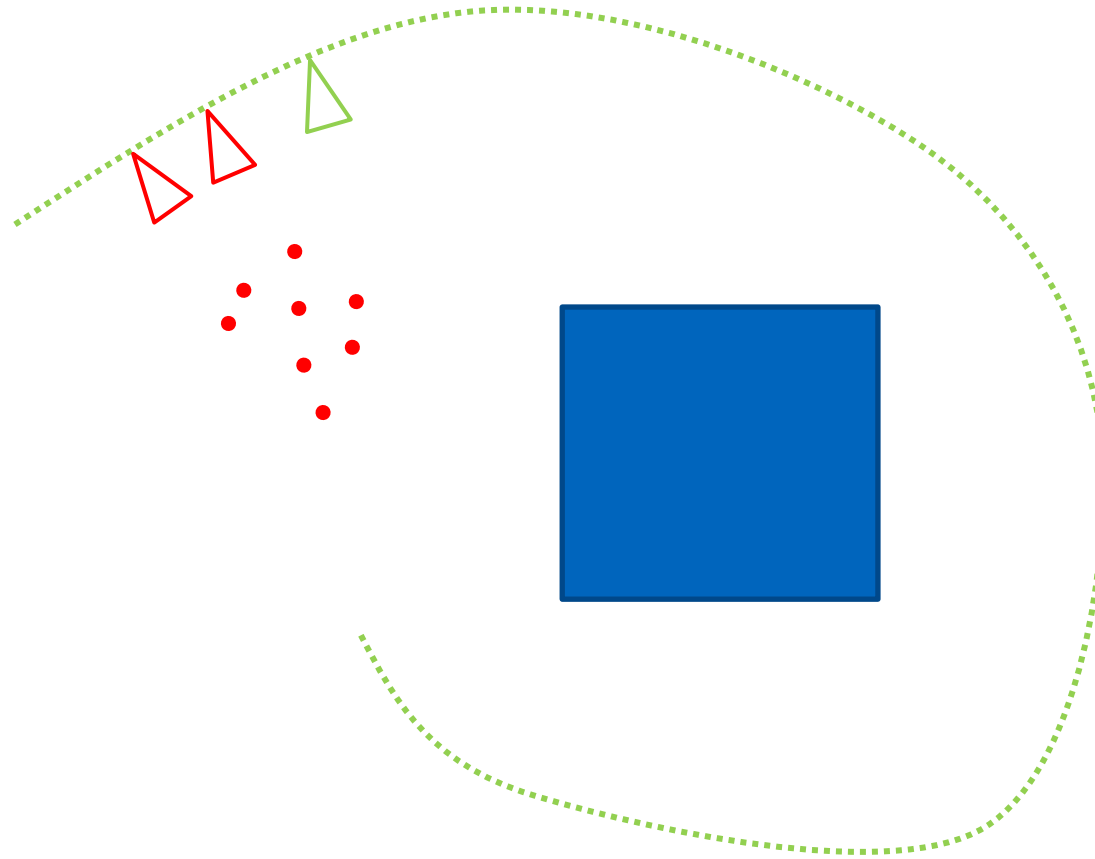
$$\mathbf{T}_{wc}^* = \operatorname{argmin}_{\mathbf{T}_{wc}} \sum_i \left\| \pi(\mathbf{T}_{wc}^{-1} \cdot \mathbf{x}_i^w) - \mathbf{u}_i \right\|^2$$



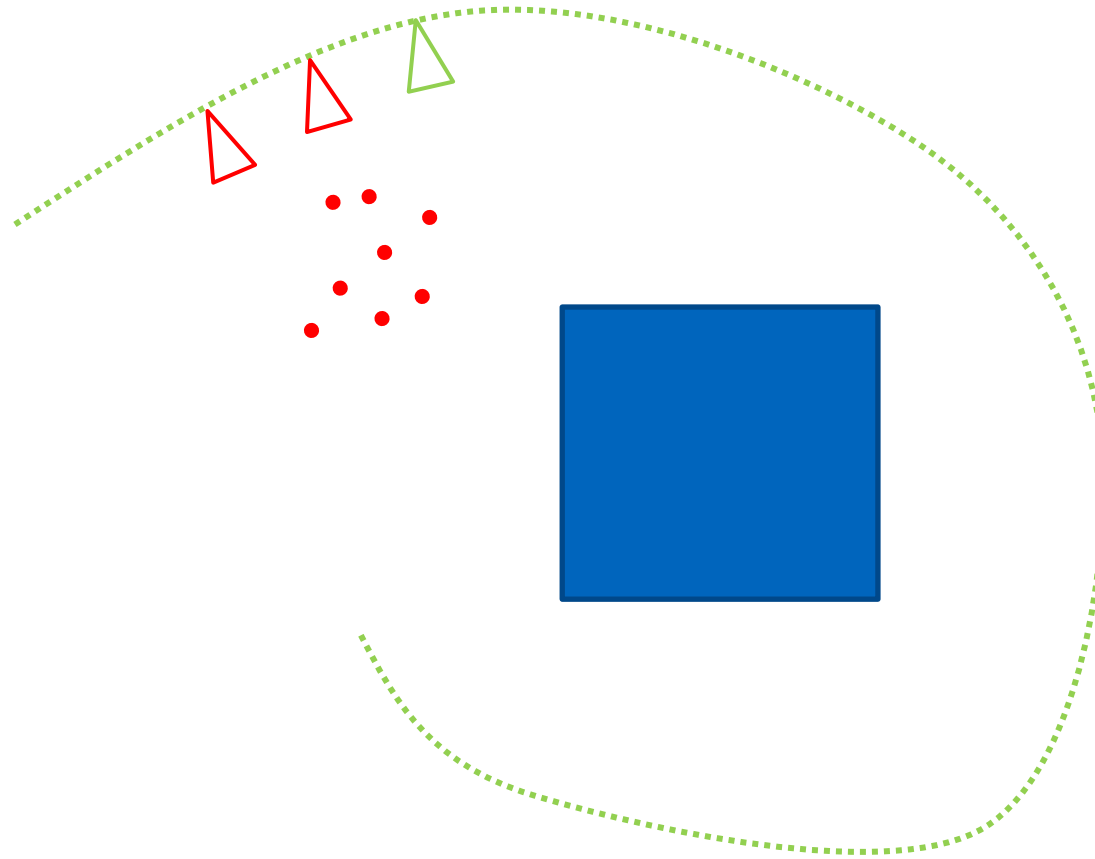
Map initialisation and tracking



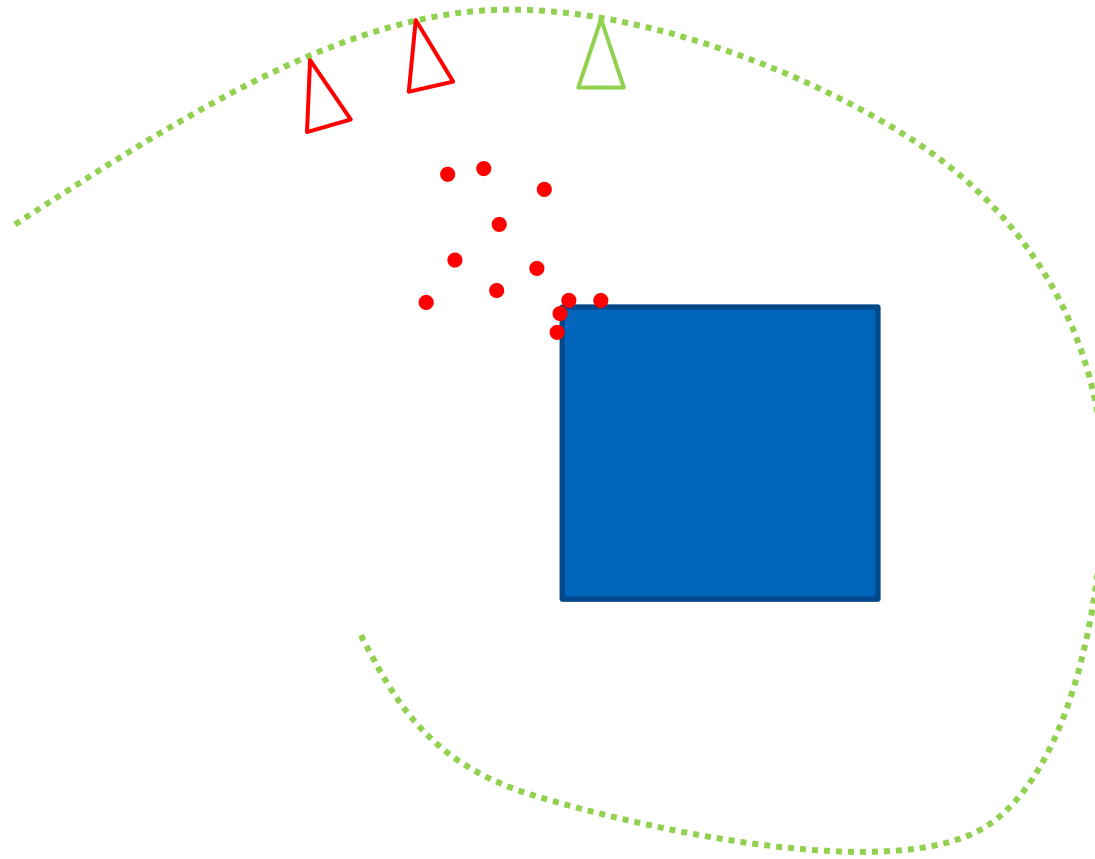
Map reinitialisation and tracking



Map reinitialisation and tracking

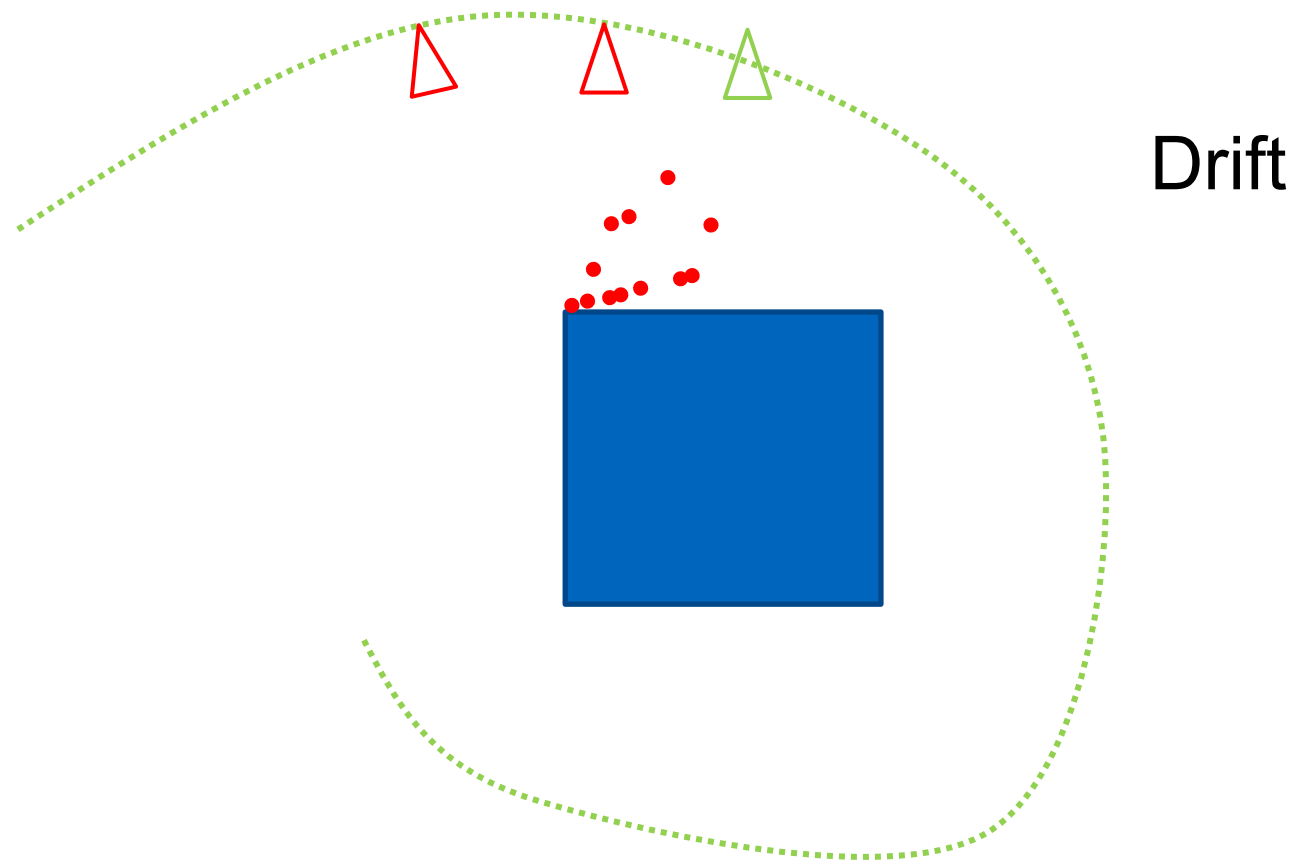


Map reinitialisation and tracking



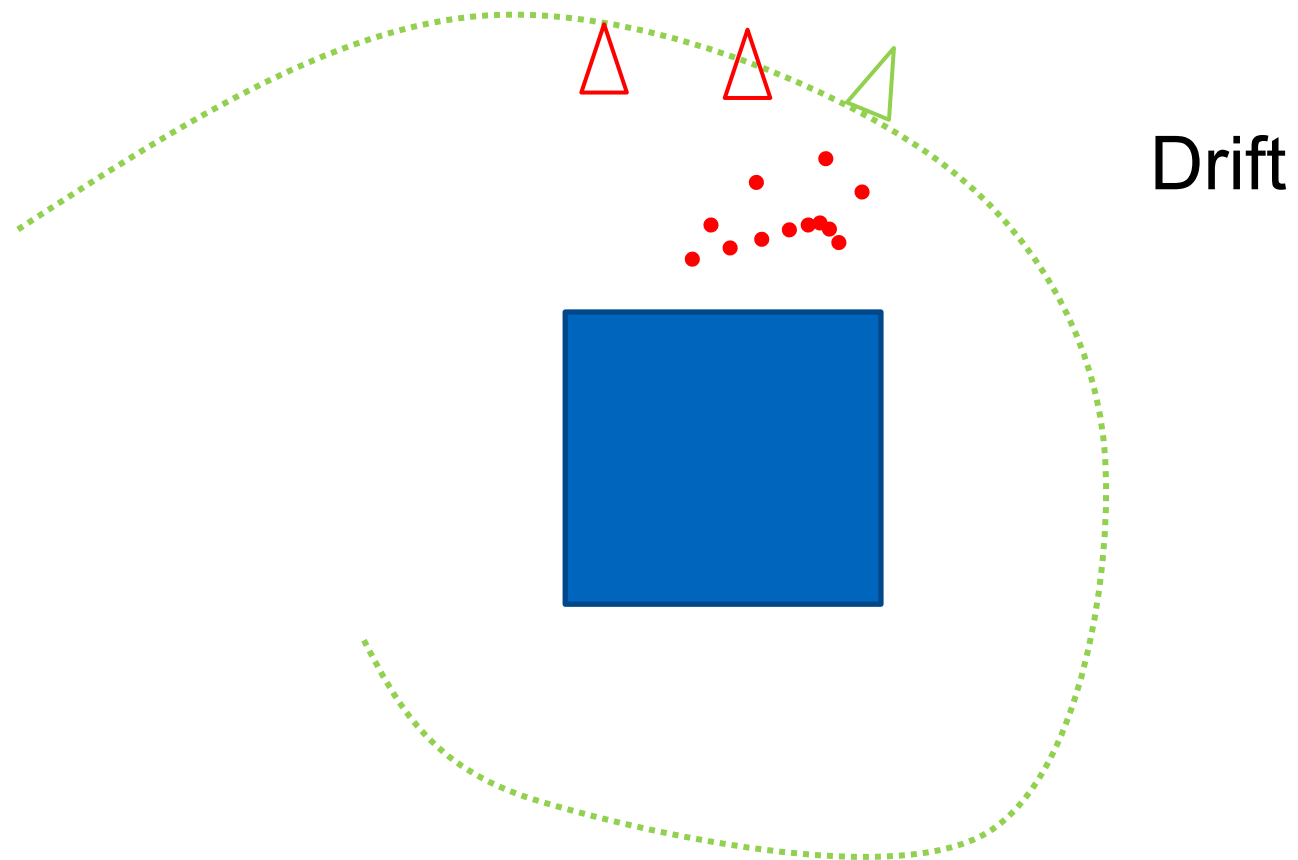
TEK5030

Map reinitialisation and tracking



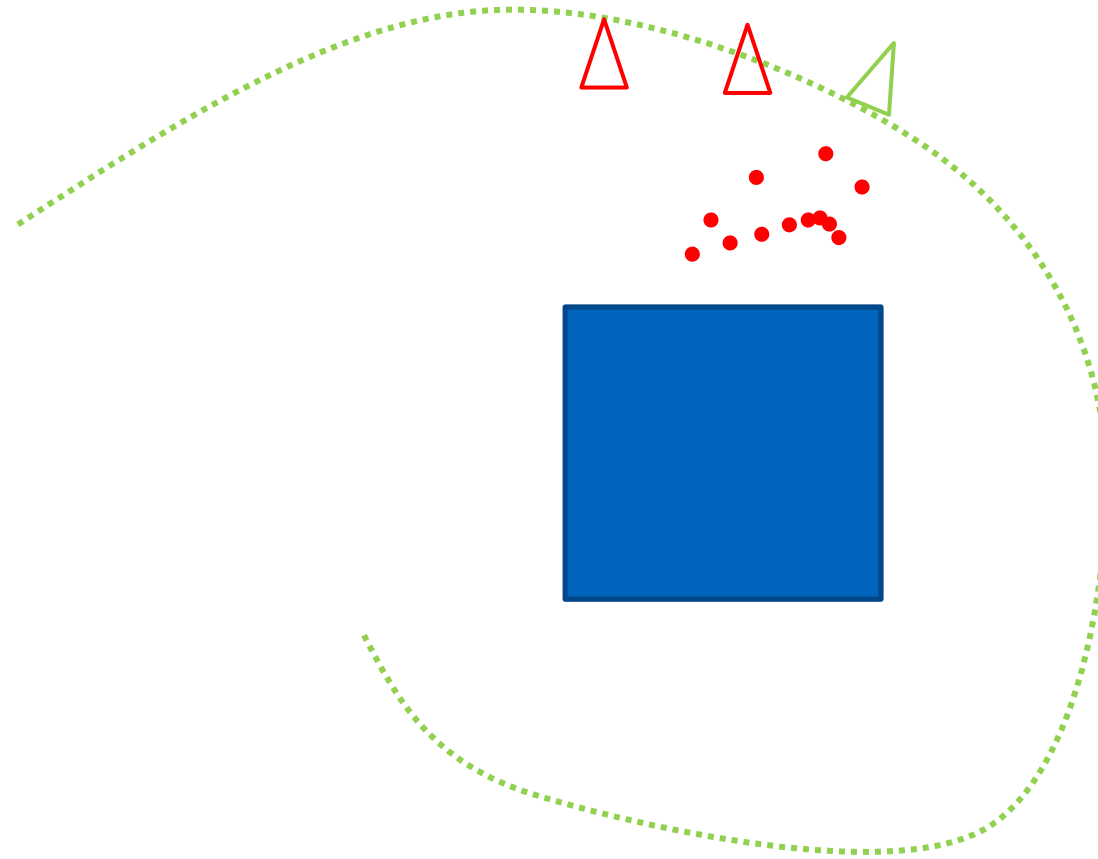
TEK5030

Map reinitialisation and tracking



TEK5030

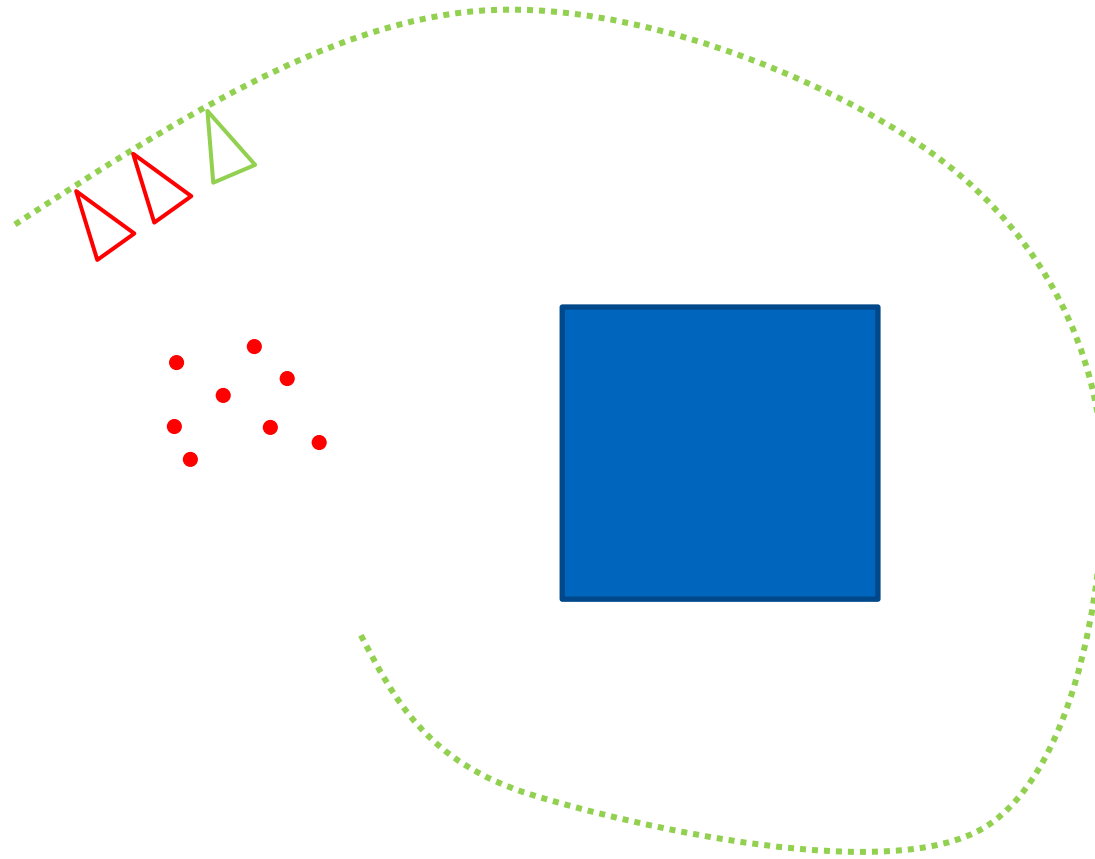
Map reinitialisation and tracking



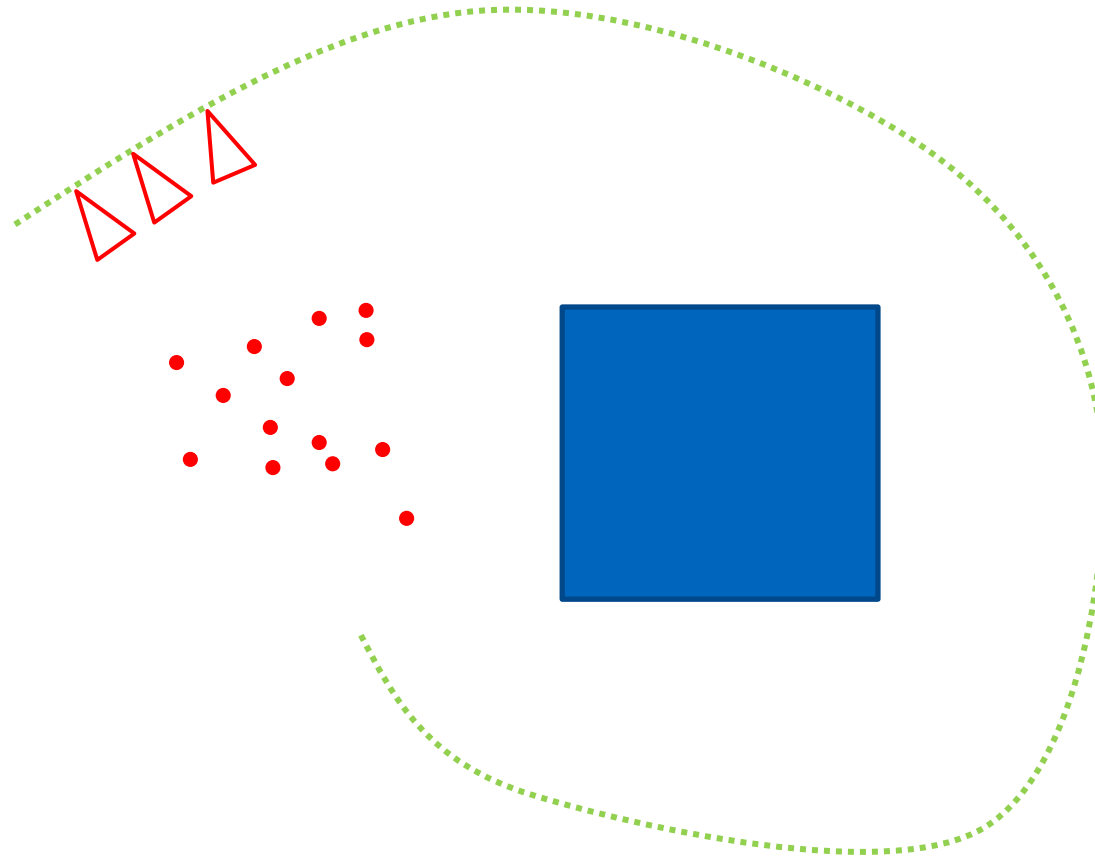
Drift

Very naïve
Visual Odometry (VO)

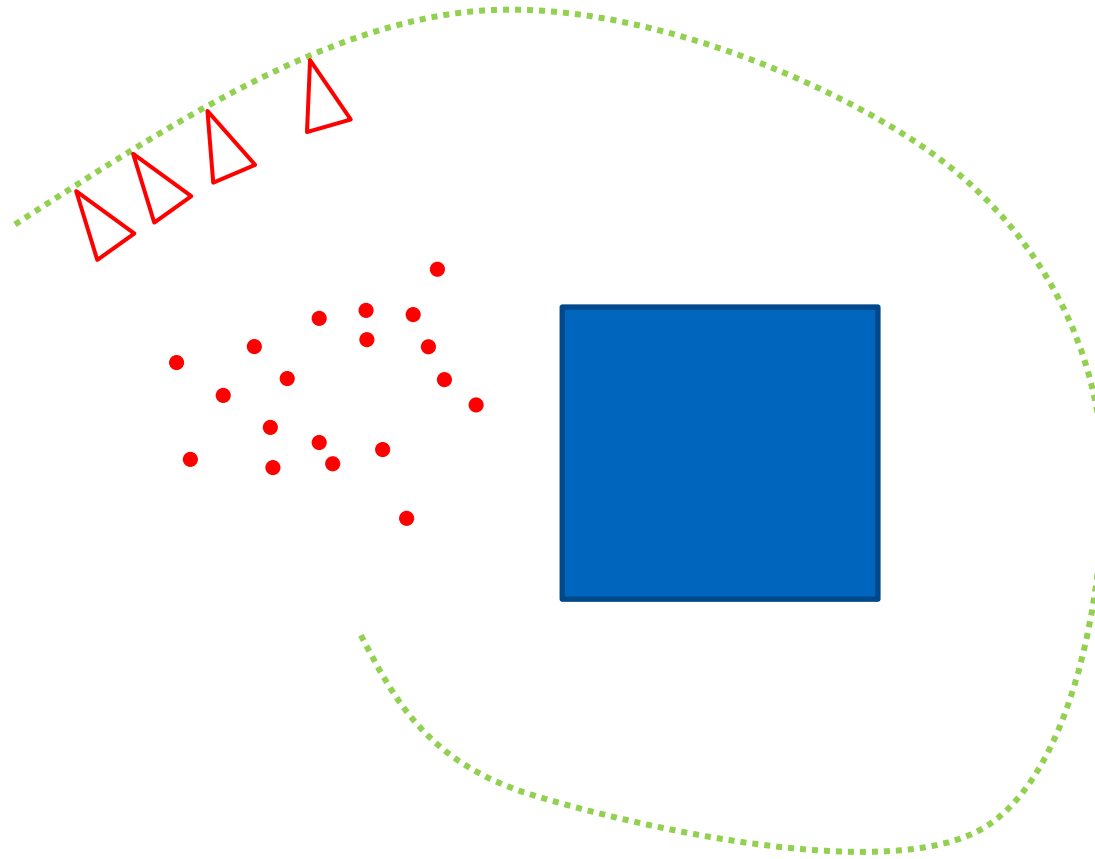
Multi-view mapping



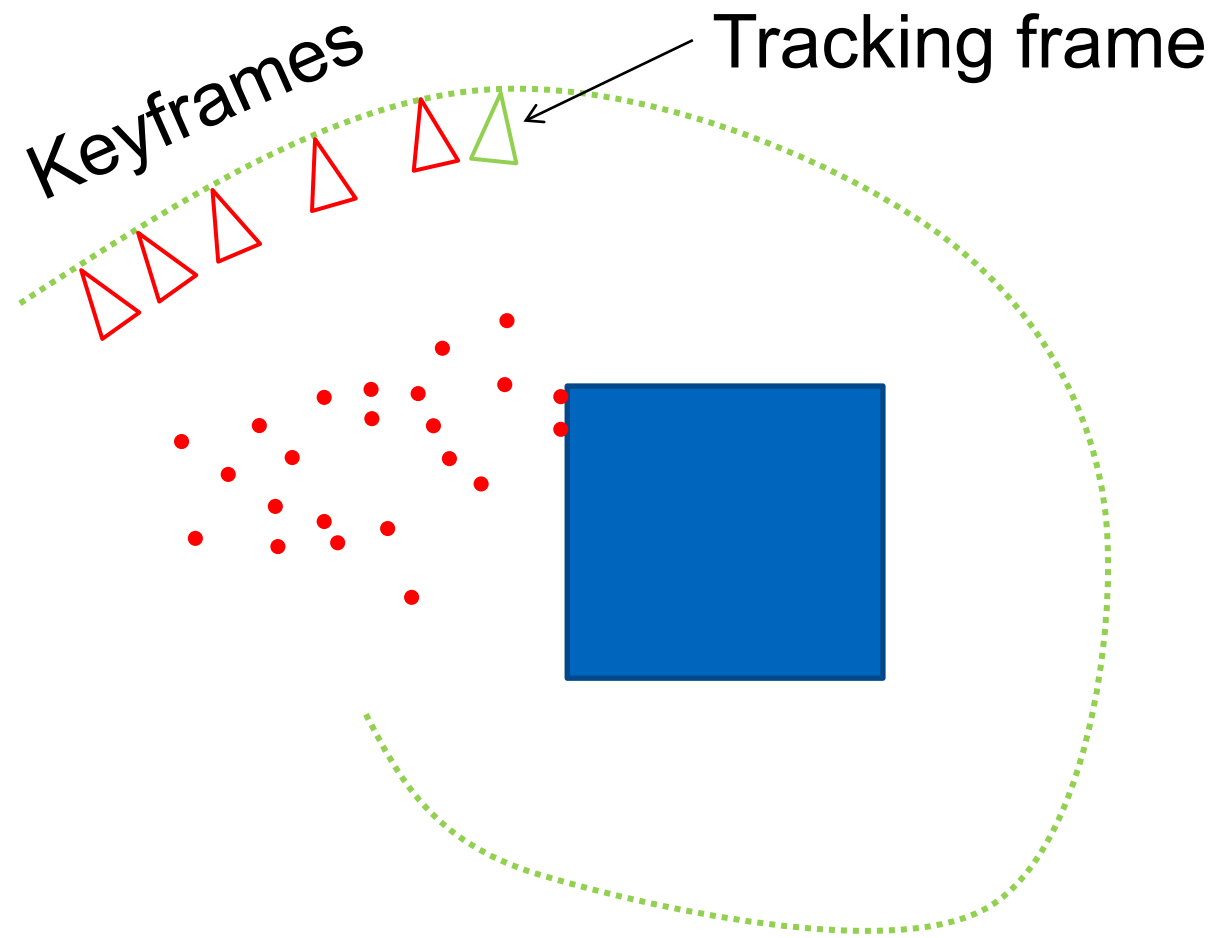
Multi-view mapping



Multi-view mapping



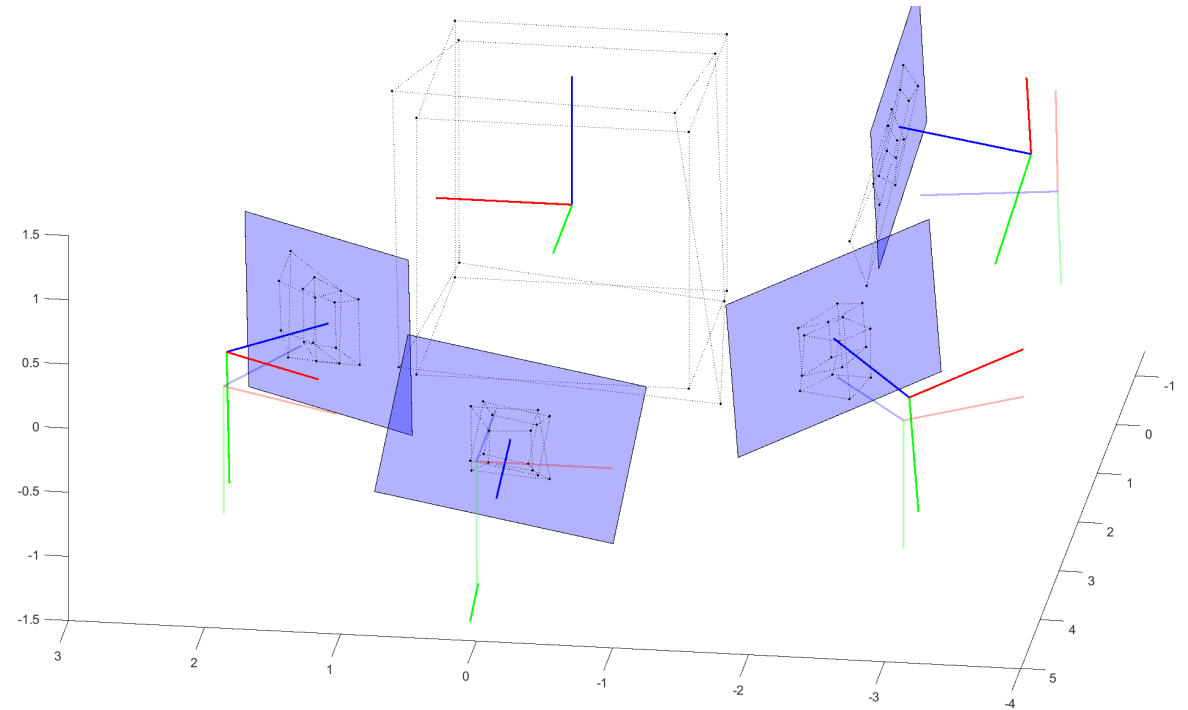
Multi-view mapping



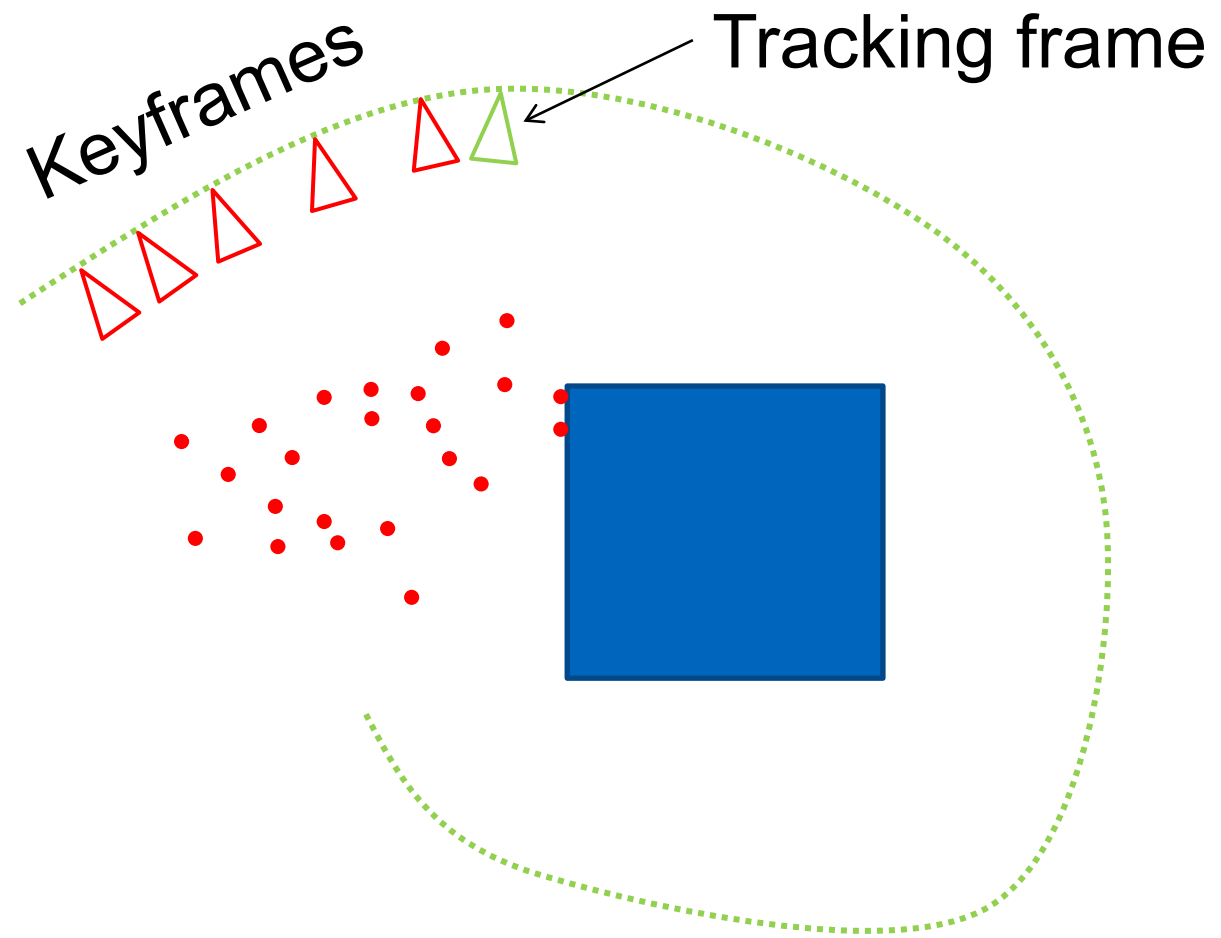
Full bundle adjustment

Minimise **geometric error** over the **camera poses** and **world points**

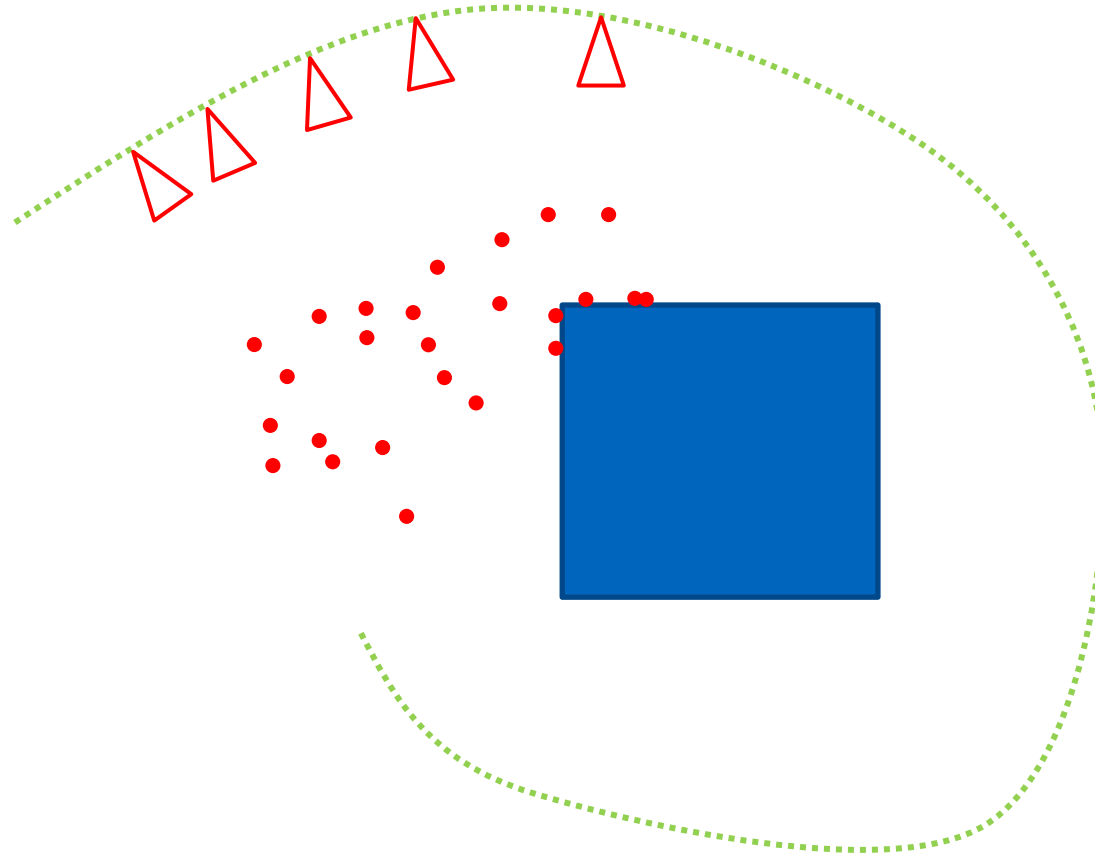
$$\left\{ \mathbf{T}_{wc_i}^*, \mathbf{x}_j^{w*} \right\} = \operatorname{argmin}_{\mathbf{T}_{wc_i}, \mathbf{x}_j^w} \sum_i \sum_j \left\| \pi_i(\mathbf{T}_{wc}^{-1} \cdot \mathbf{x}_j^w) - \mathbf{u}_j^i \right\|^2$$



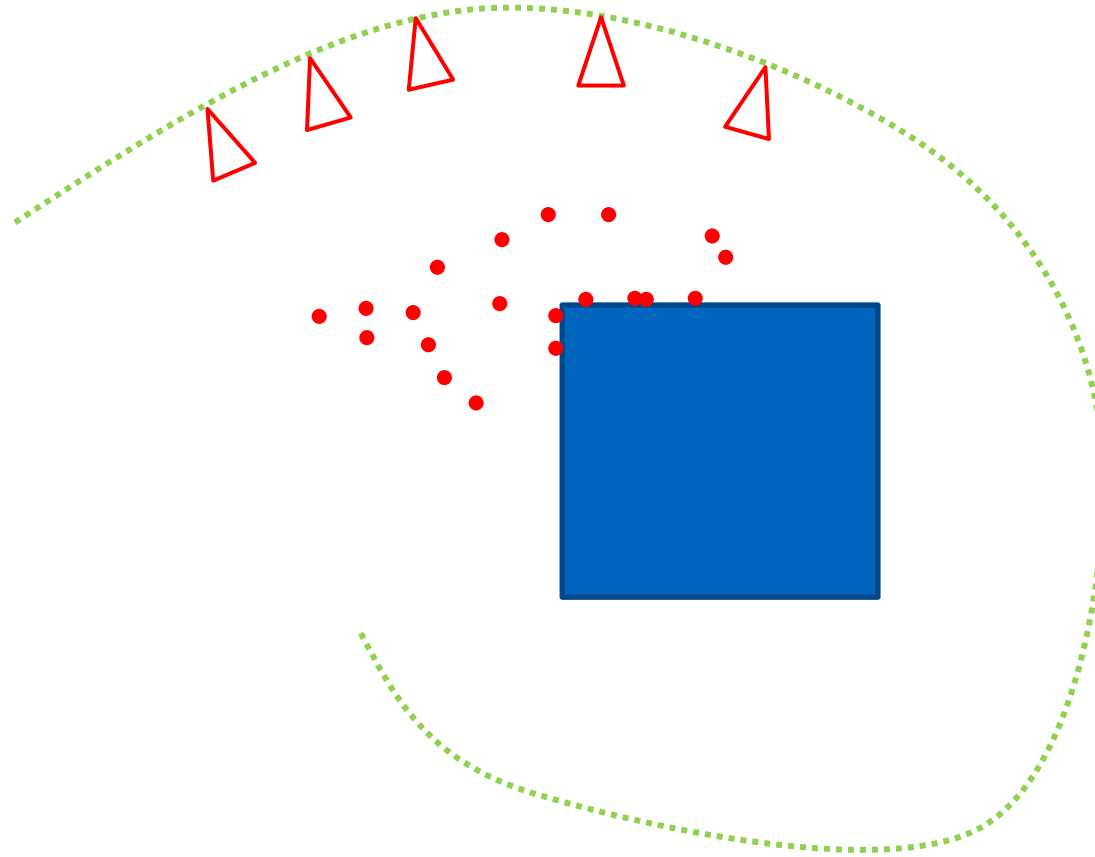
Multi-view mapping



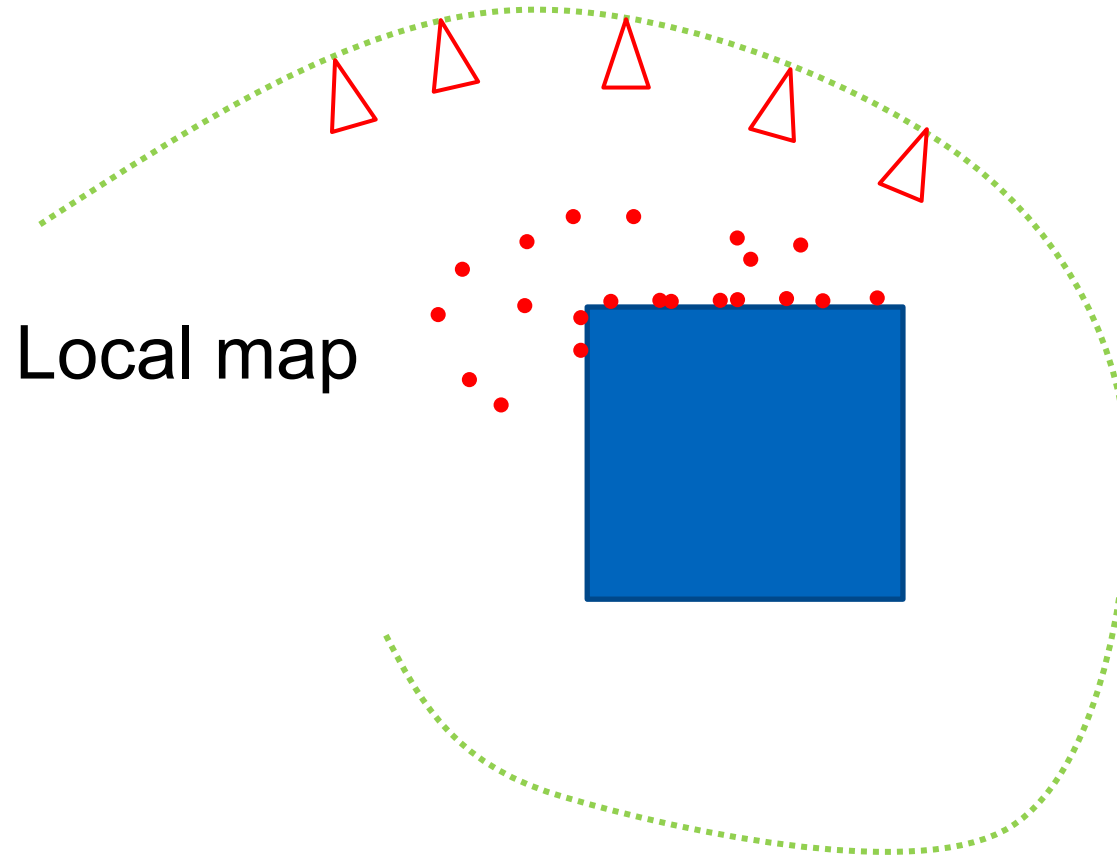
Sliding window mapping



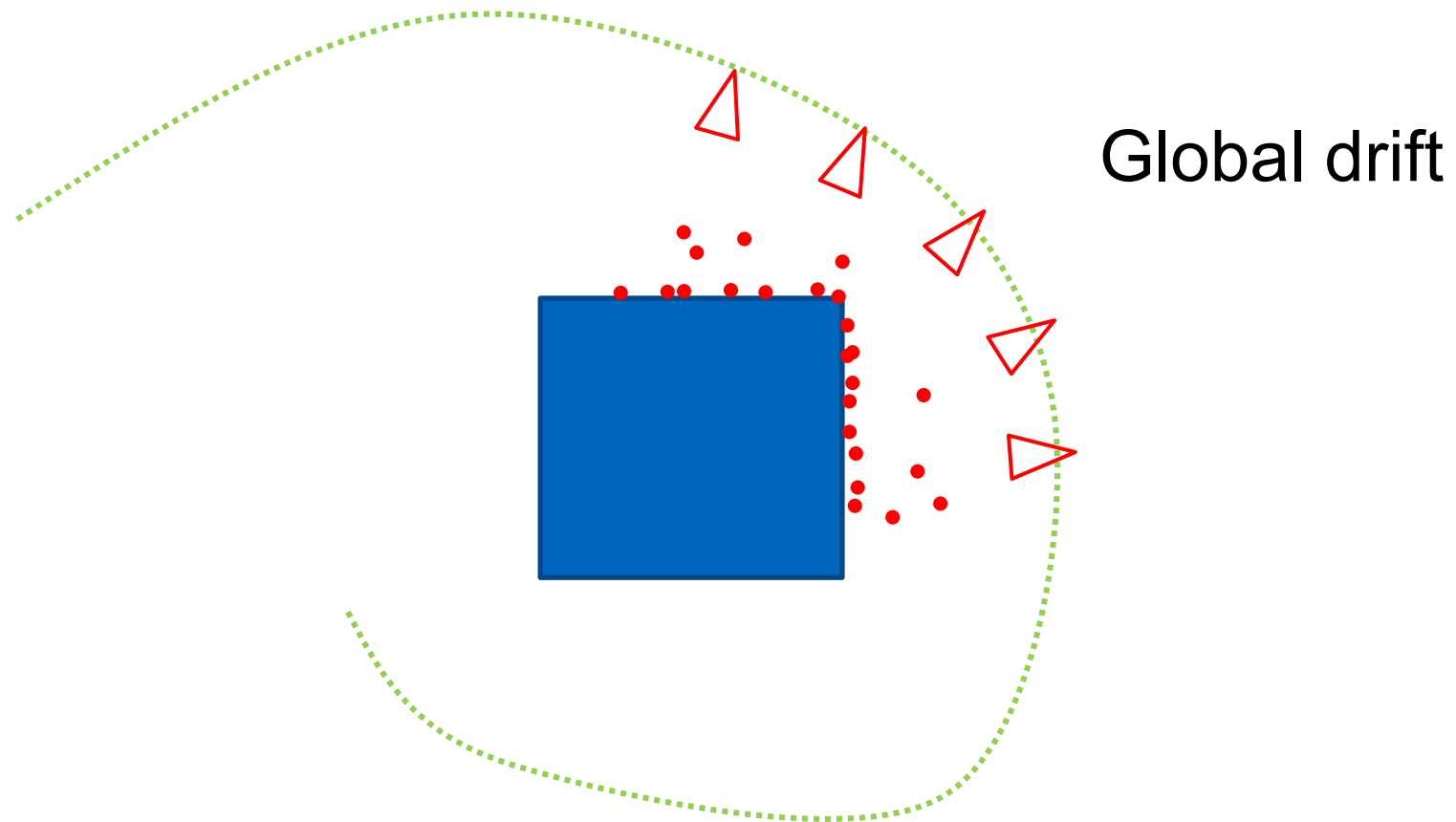
Sliding window mapping



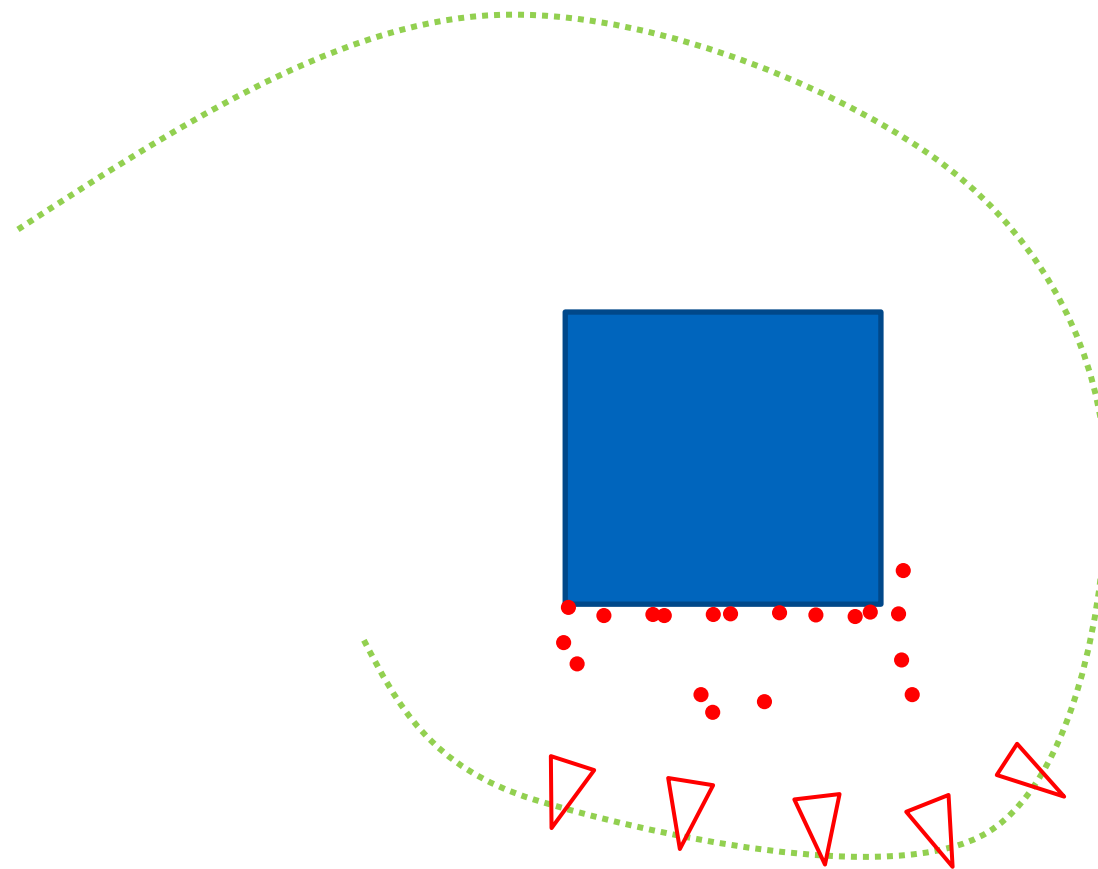
Sliding window mapping



Sliding window mapping



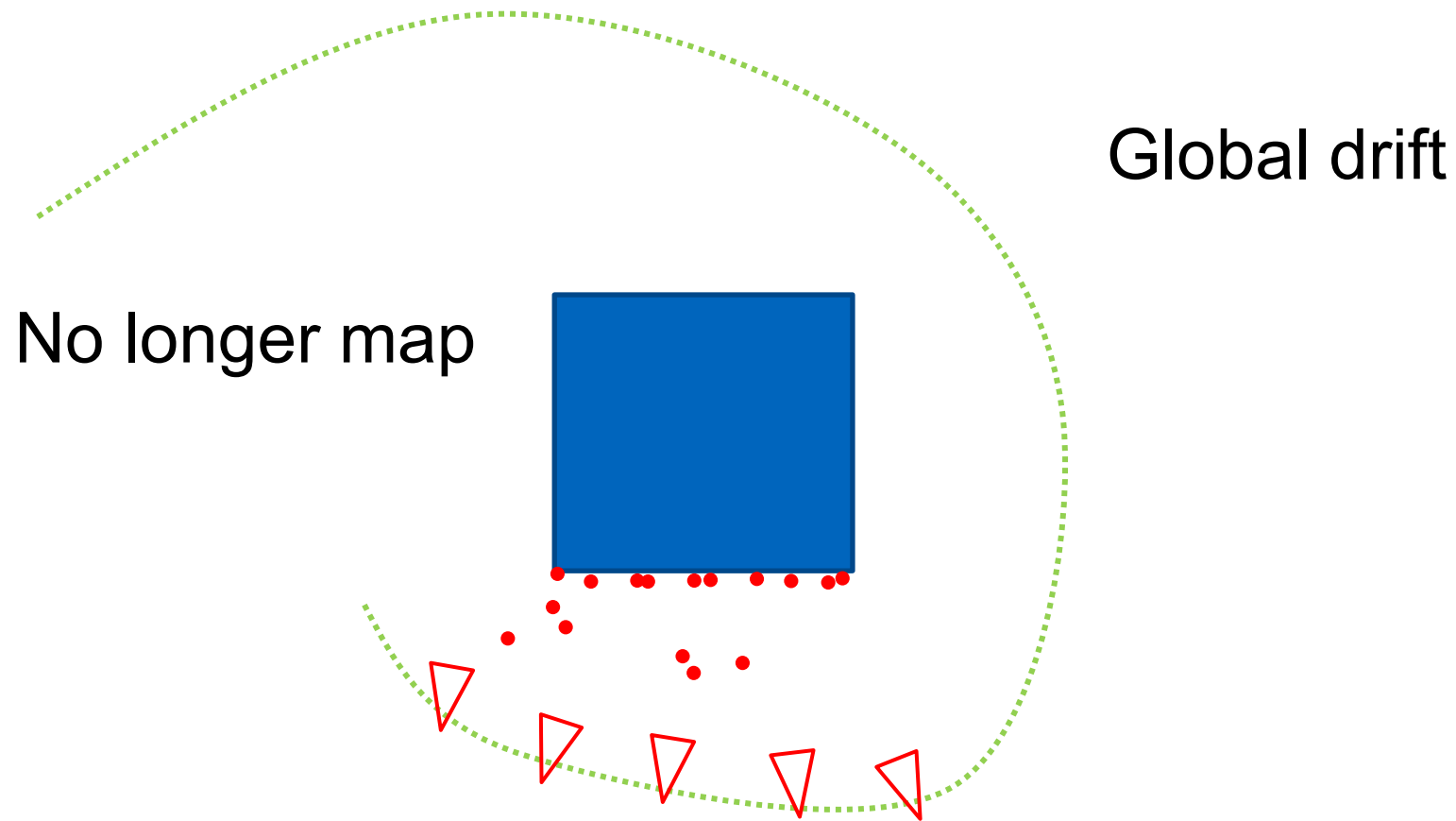
Sliding window mapping



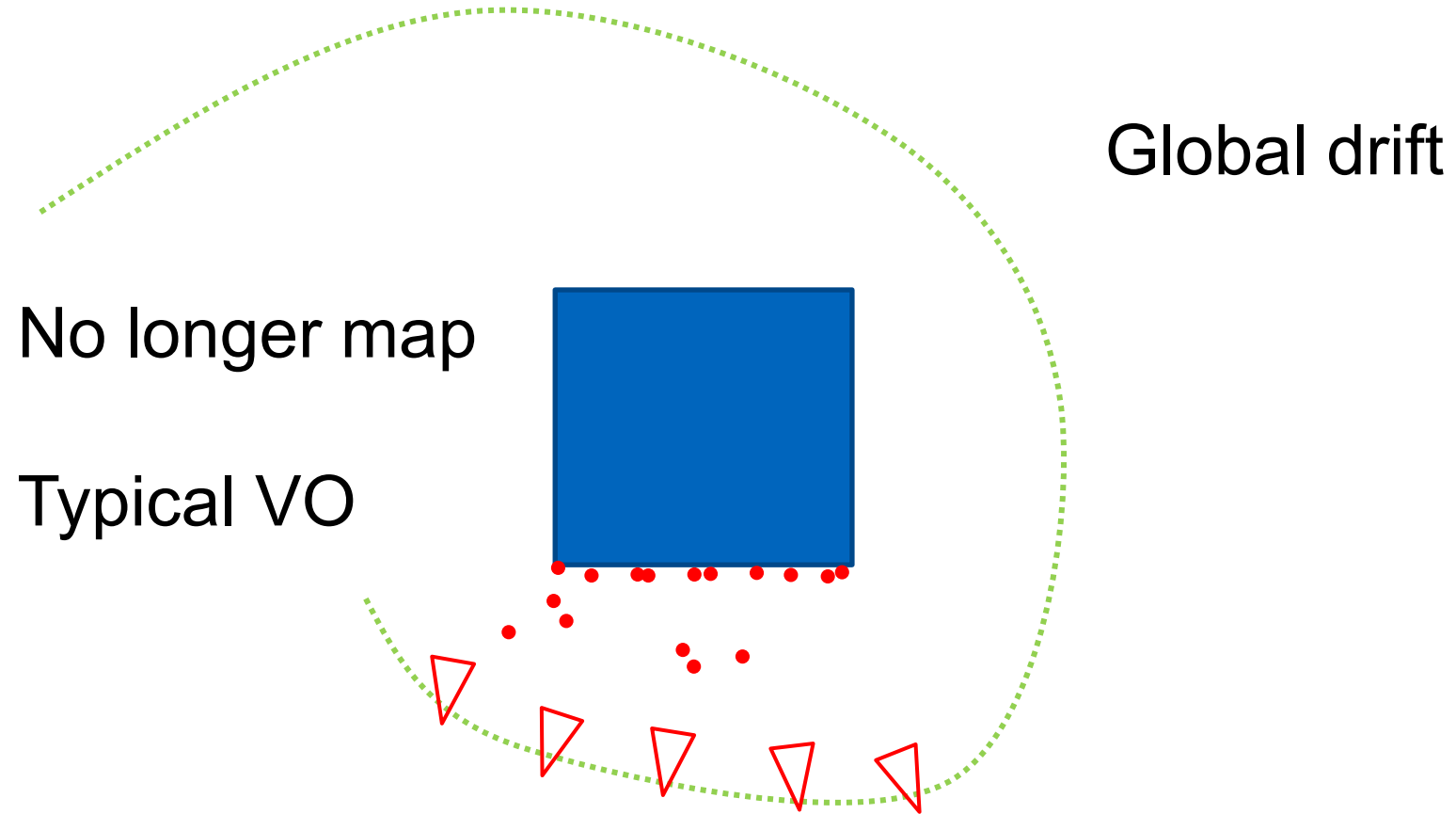
Global drift

TEK5030

Sliding window mapping

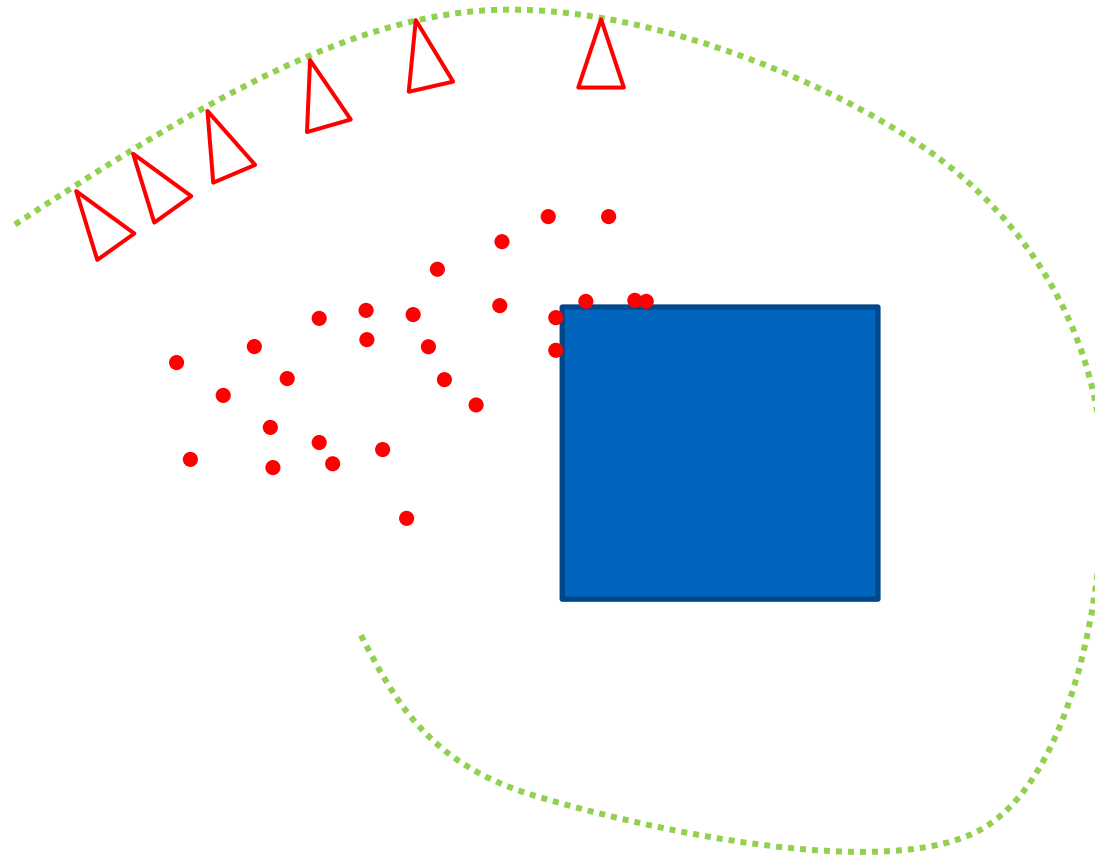


Sliding window mapping

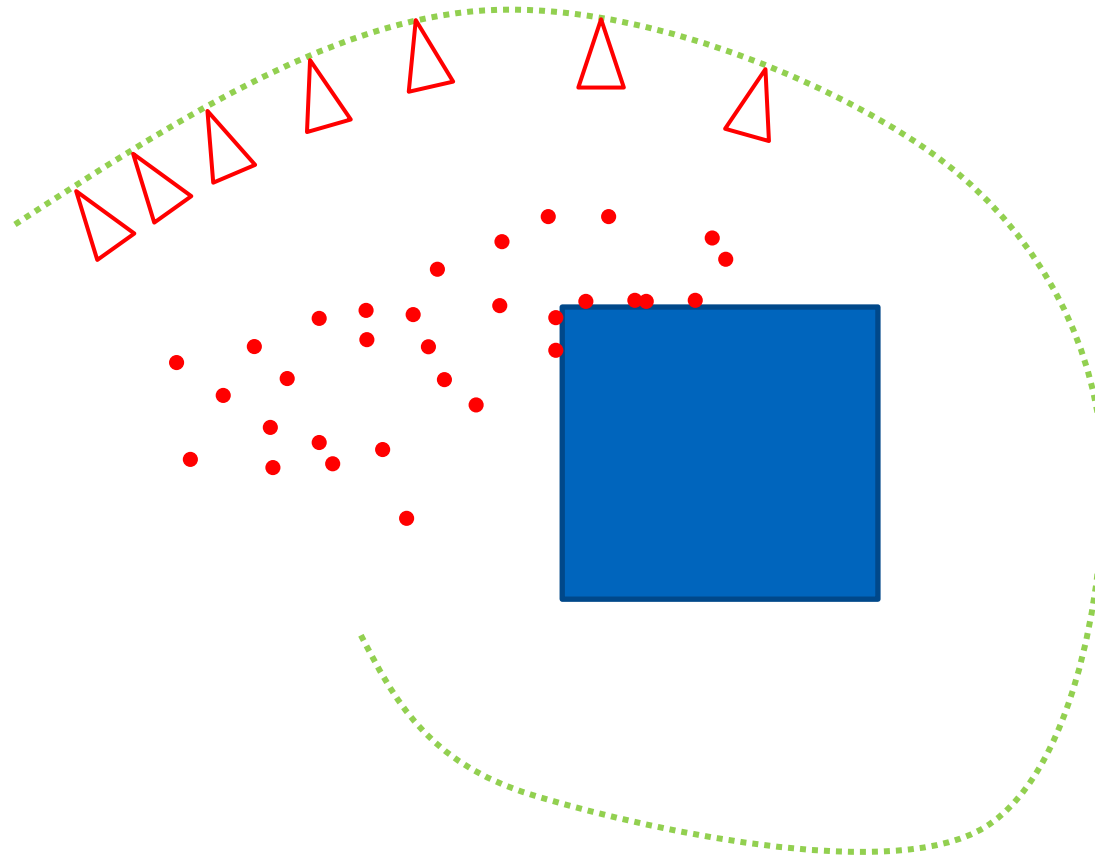


TEK5030

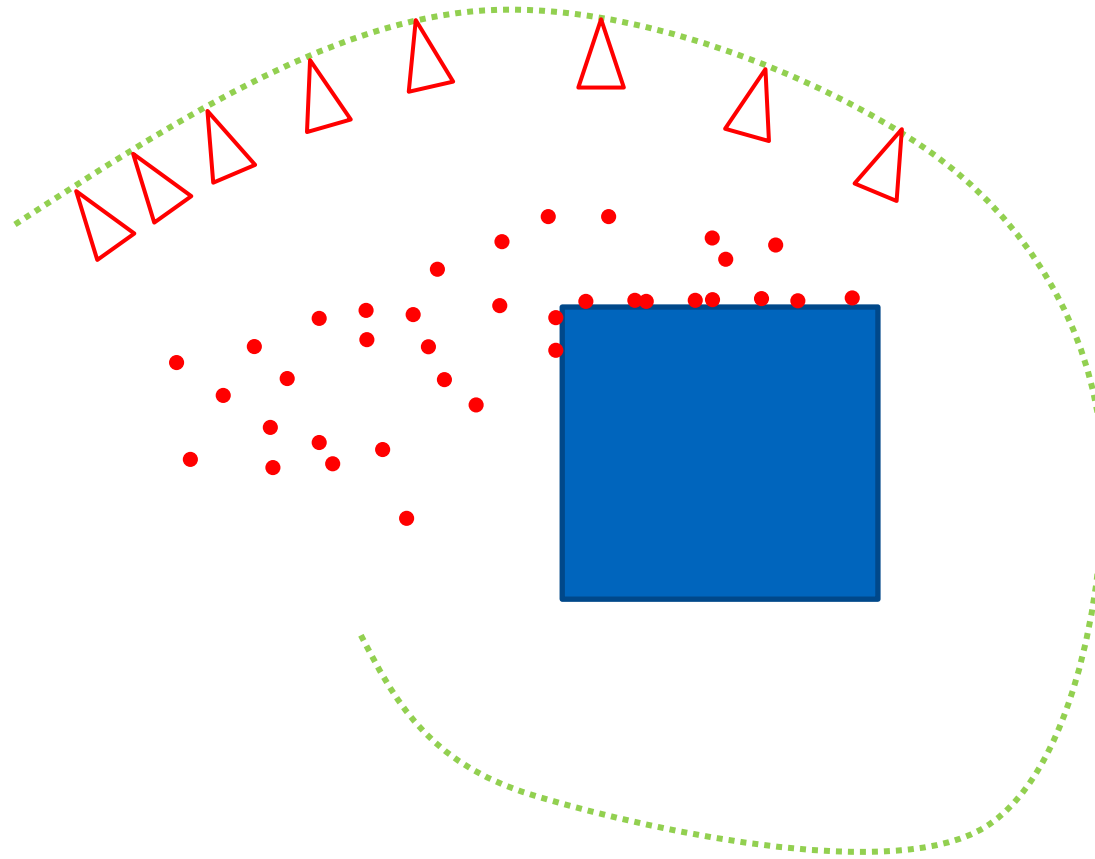
Monocular Visual SLAM



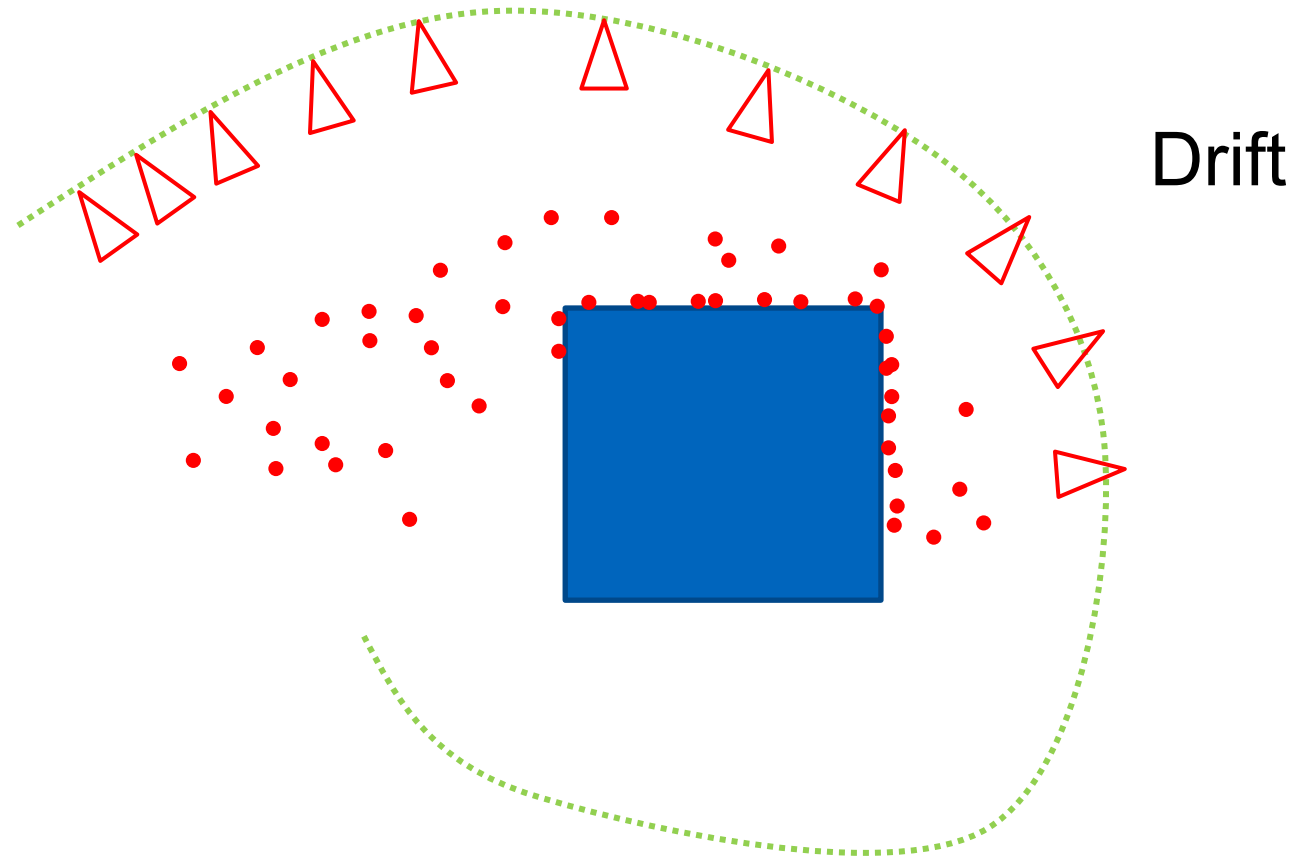
Monocular Visual SLAM



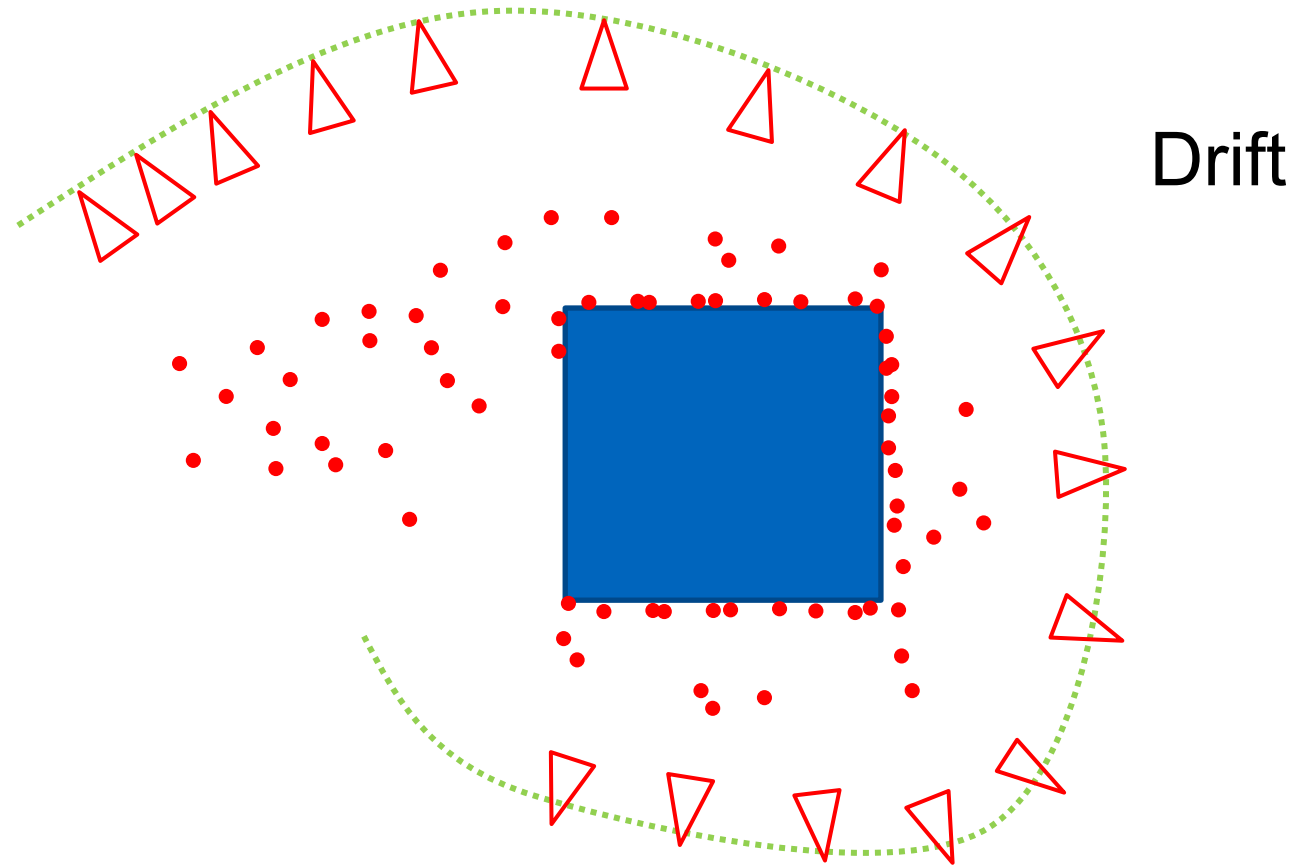
Monocular Visual SLAM



Monocular Visual SLAM

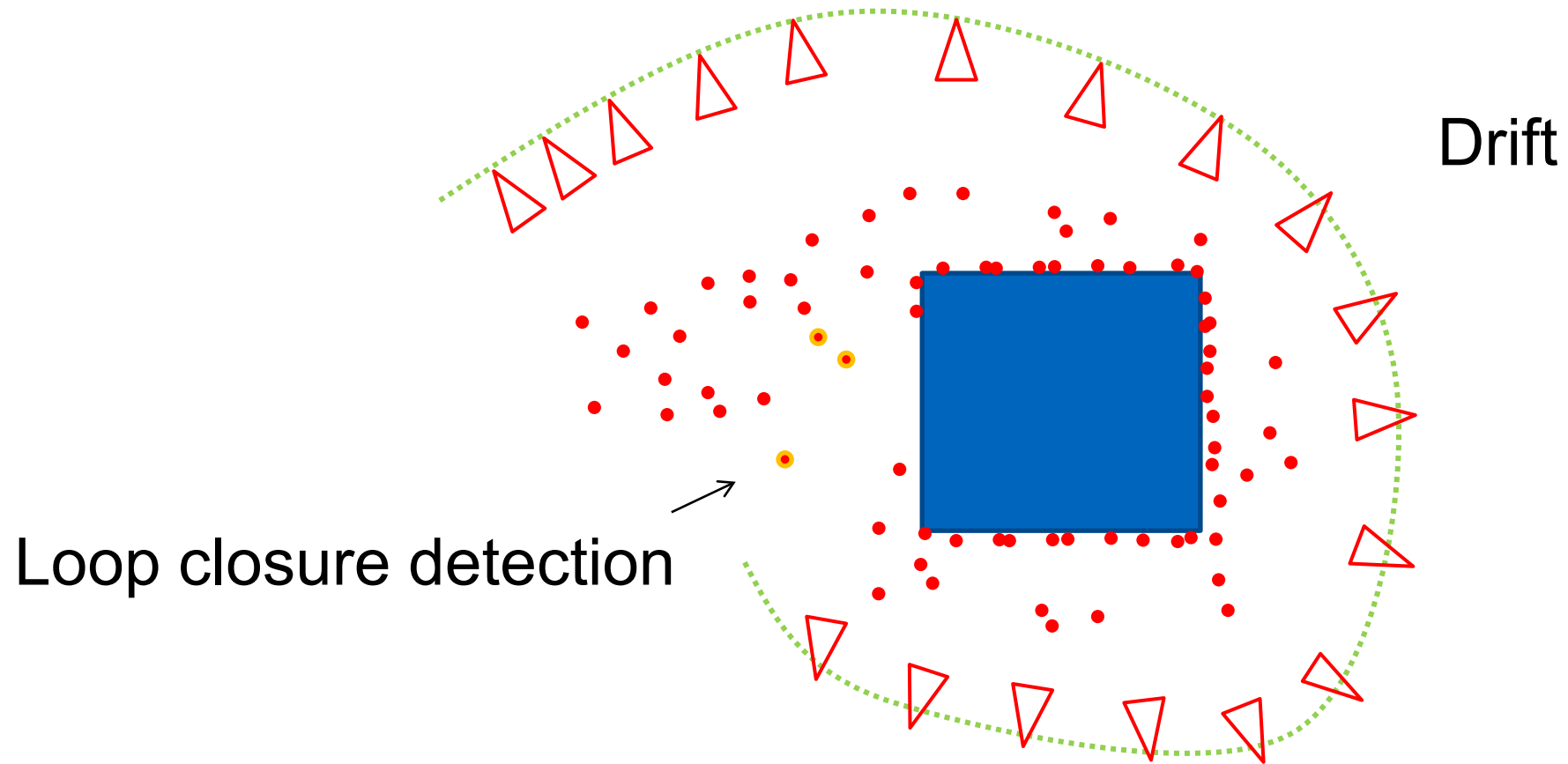


Monocular Visual SLAM

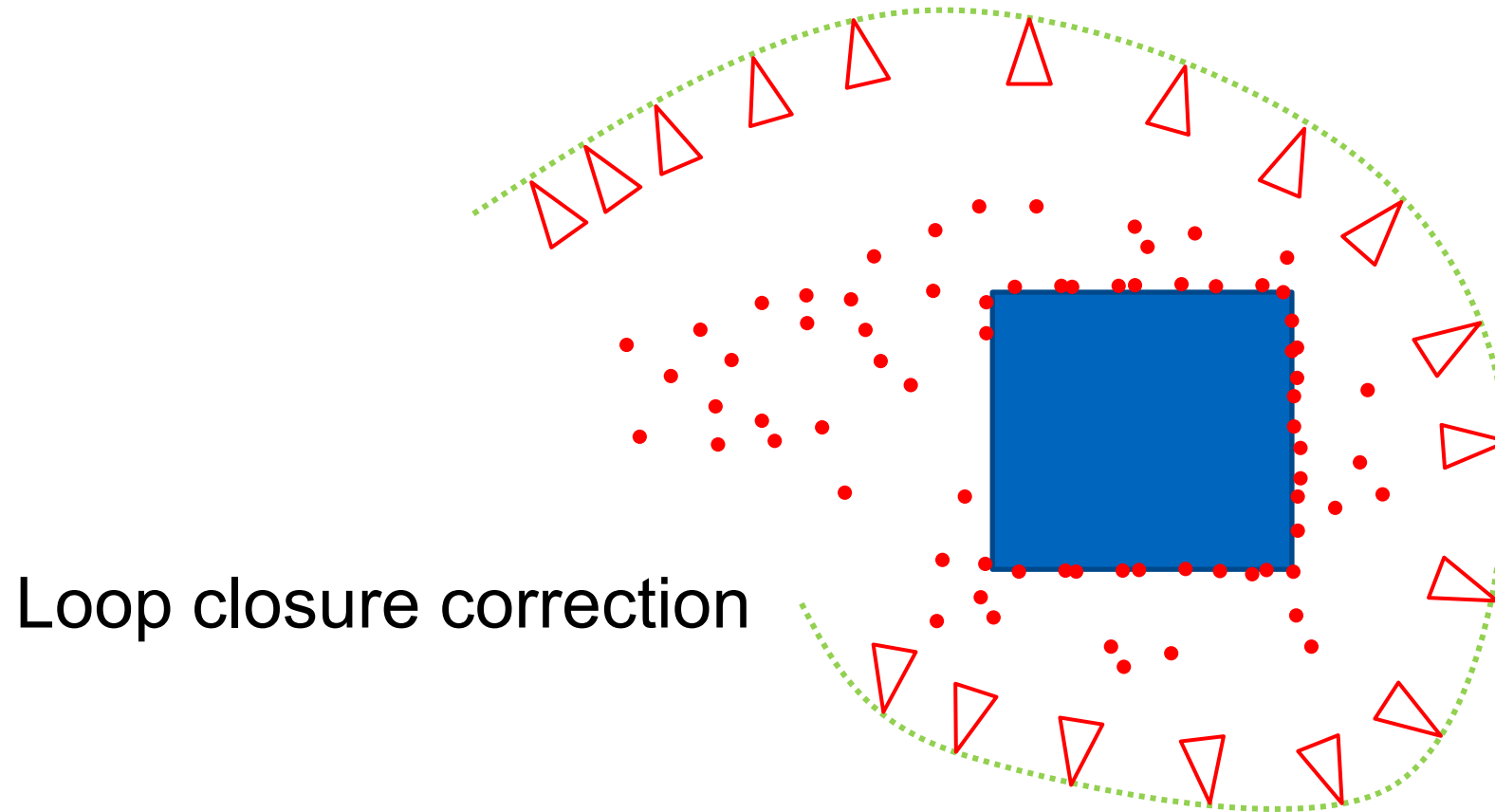


TEK5030

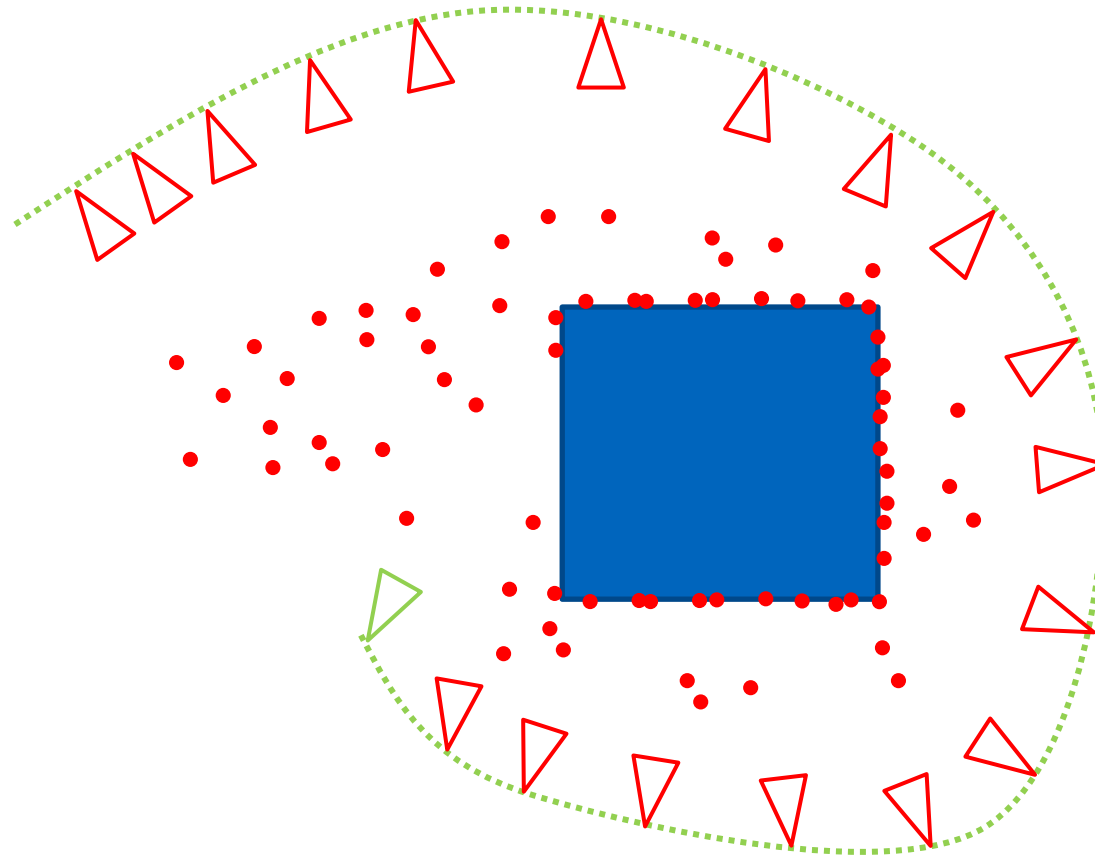
Monocular Visual SLAM



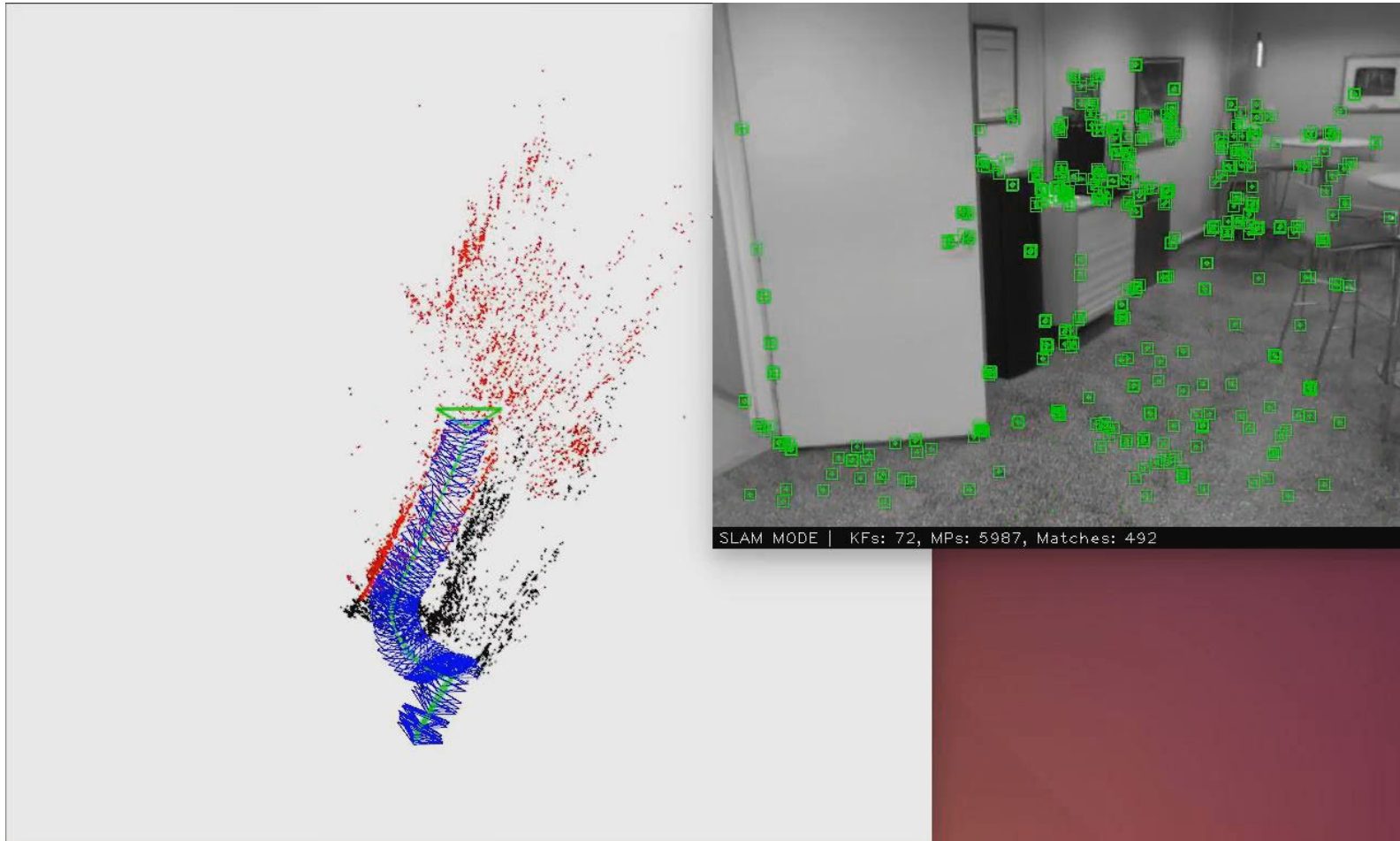
Monocular Visual SLAM



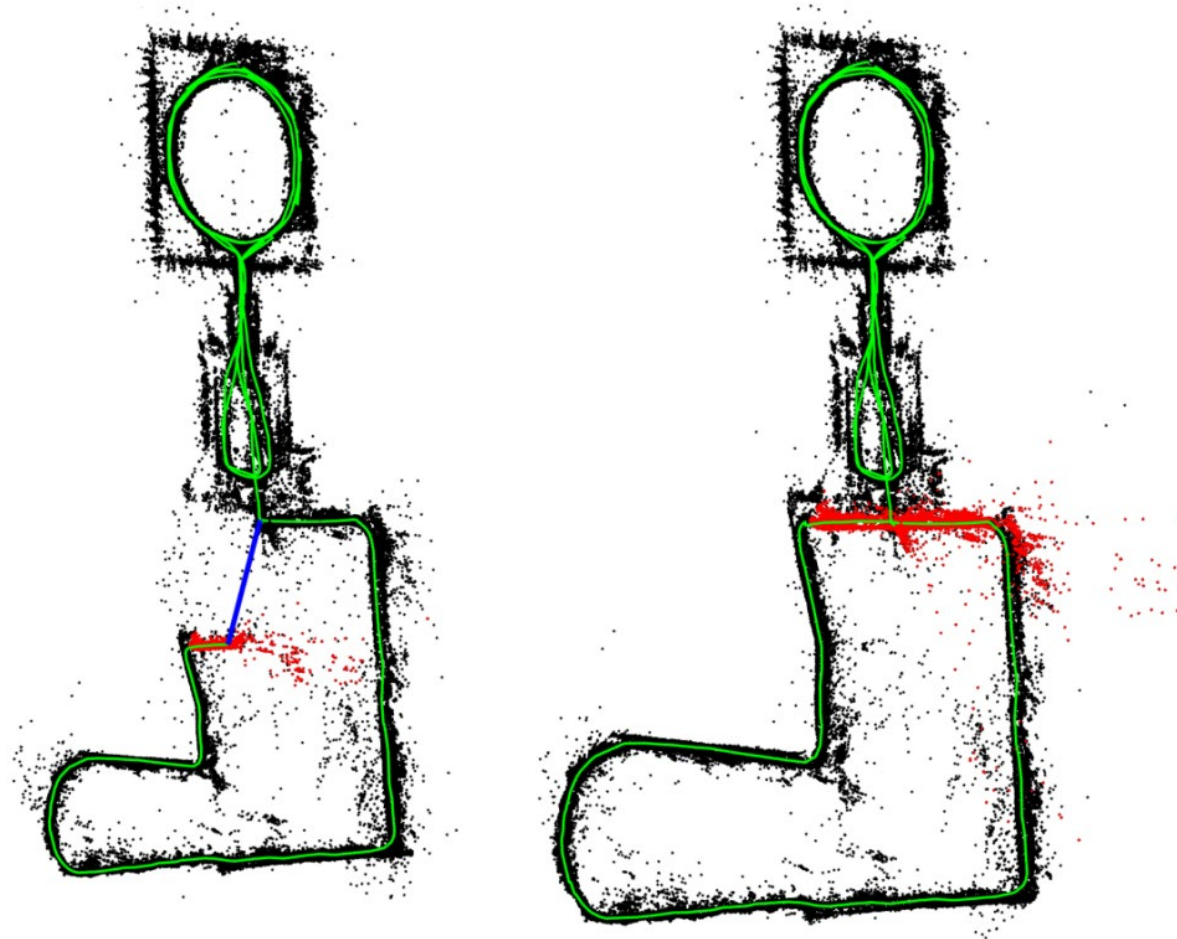
Monocular Visual SLAM



Visual SLAM example

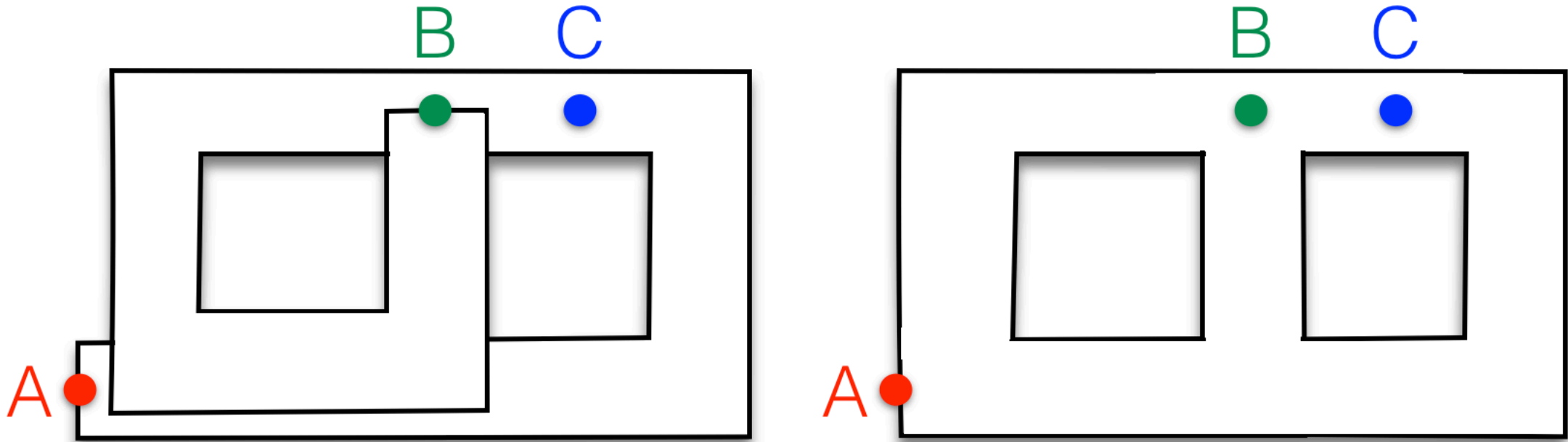


Visual SLAM vs visual odometry



Mur-Artal, R., Montiel, J. M. M., & Tardos, J. D. (2015). ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*, 31(5), 1147–1163

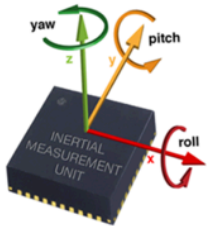
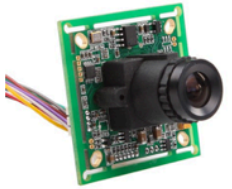
Visual SLAM vs visual odometry



Cadena, C., et al. (2016). Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age. *IEEE Transactions on Robotics*, 32(6), 1309–1332

Components of SLAM

sensor
data



front-end

feature extraction

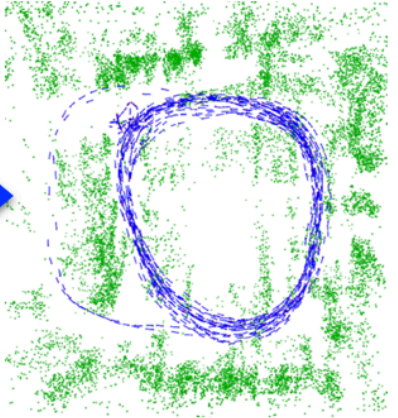
data association:

- short-term (feature tracking)
- long-term (loop closure)



back-end

MAP
estimation

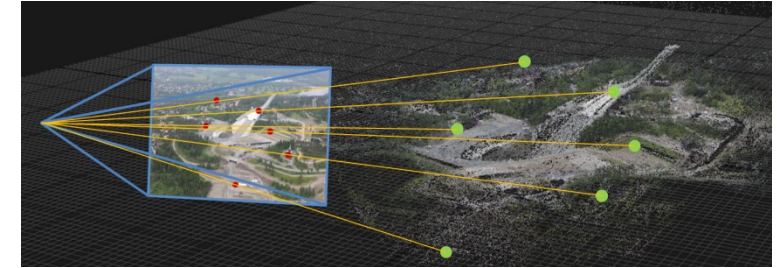


SLAM
estimate

Cadena, C., et al. (2016). Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age. *IEEE Transactions on Robotics*, 32(6), 1309–1332

Components of VSLAM

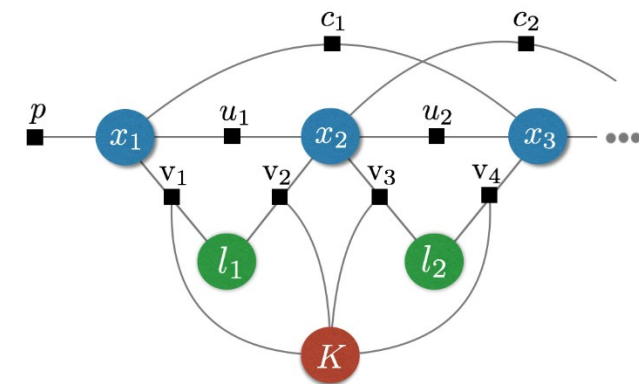
- Short-term tracking
 - Pose estimation given the local map
 - Keyframe proposals
- Mid-term tracking
 - Loop closure detection in the local map
- Long-term tracking
 - Loop closure detection in the global map
- Mapping
 - Building and optimising the map over keyframes both locally and globally
 - Data fusion



(a)

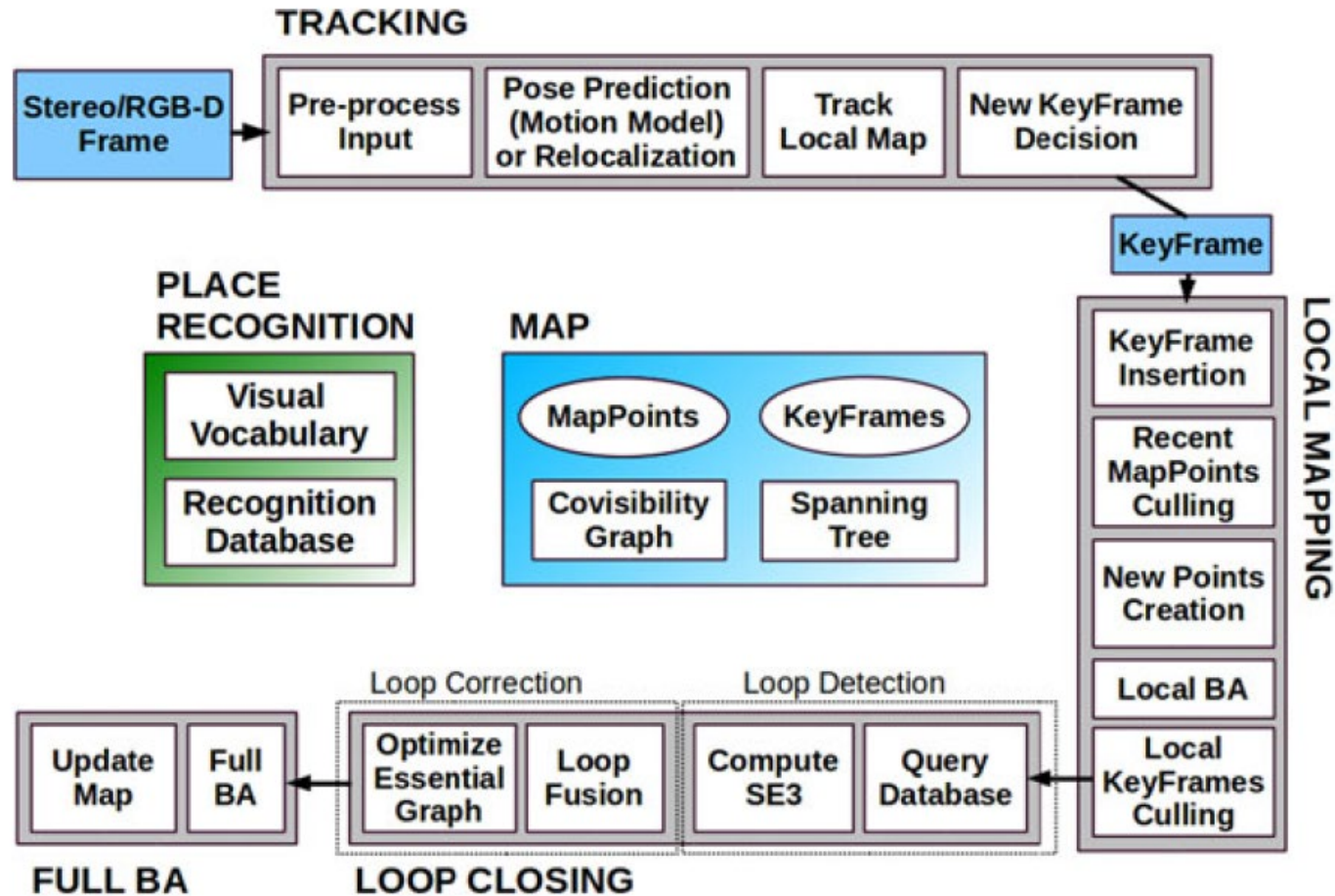


Lowry, S. et al. (2016). Visual Place Recognition: A Survey. IEEE Transactions on Robotics, 32(1), 1–19.



Cadena, C., et al. (2016). Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age. *IEEE Transactions on Robotics*, 32(6), 1309–1332

Example: ORB-SLAM 2



R. Mur-Artal and J. D. Tardos, "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras," *IEEE Trans. Robot.*, pp. 1–8, 2017.

Part II

SHORT-TERM, MID-TERM AND LONG-TERM TRACKING

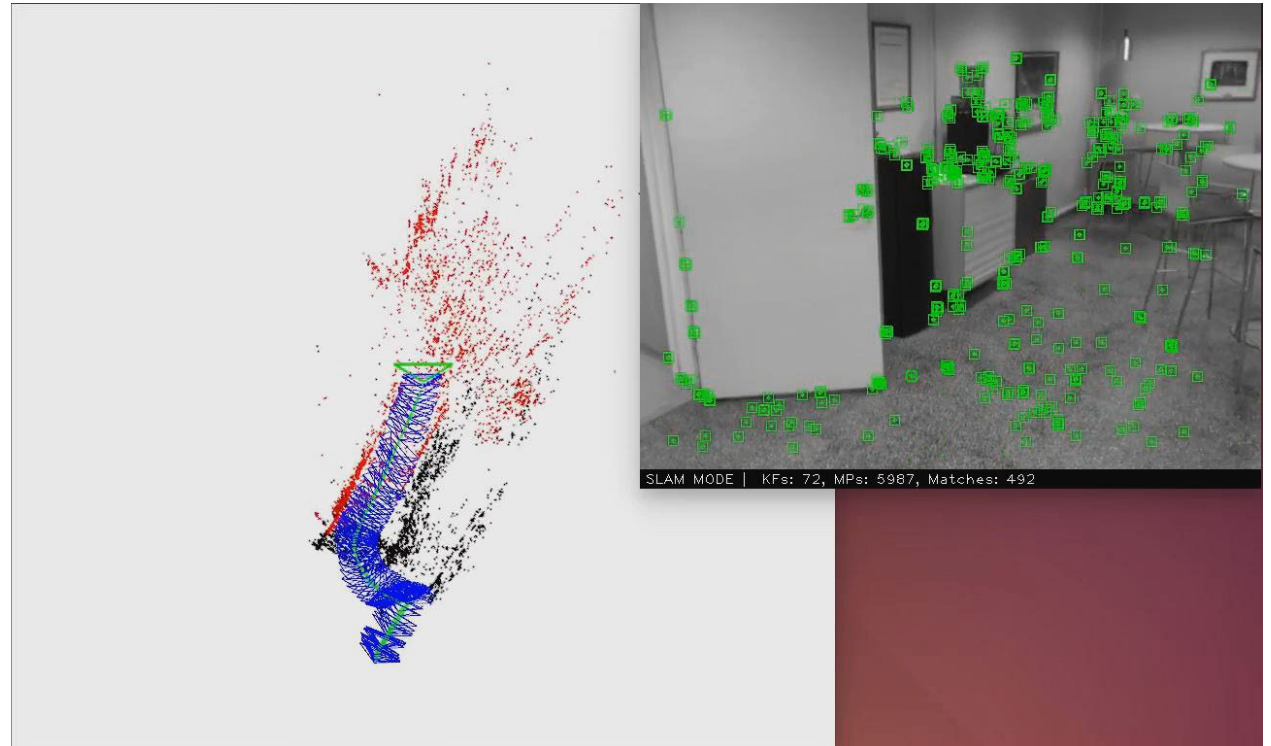
Tracking the map in VSLAM

We track the map for localisation

- Estimate the camera pose relative to the map for each frame

and for building a consistent map

- Detect loop closures



Tracking the map in VSLAM

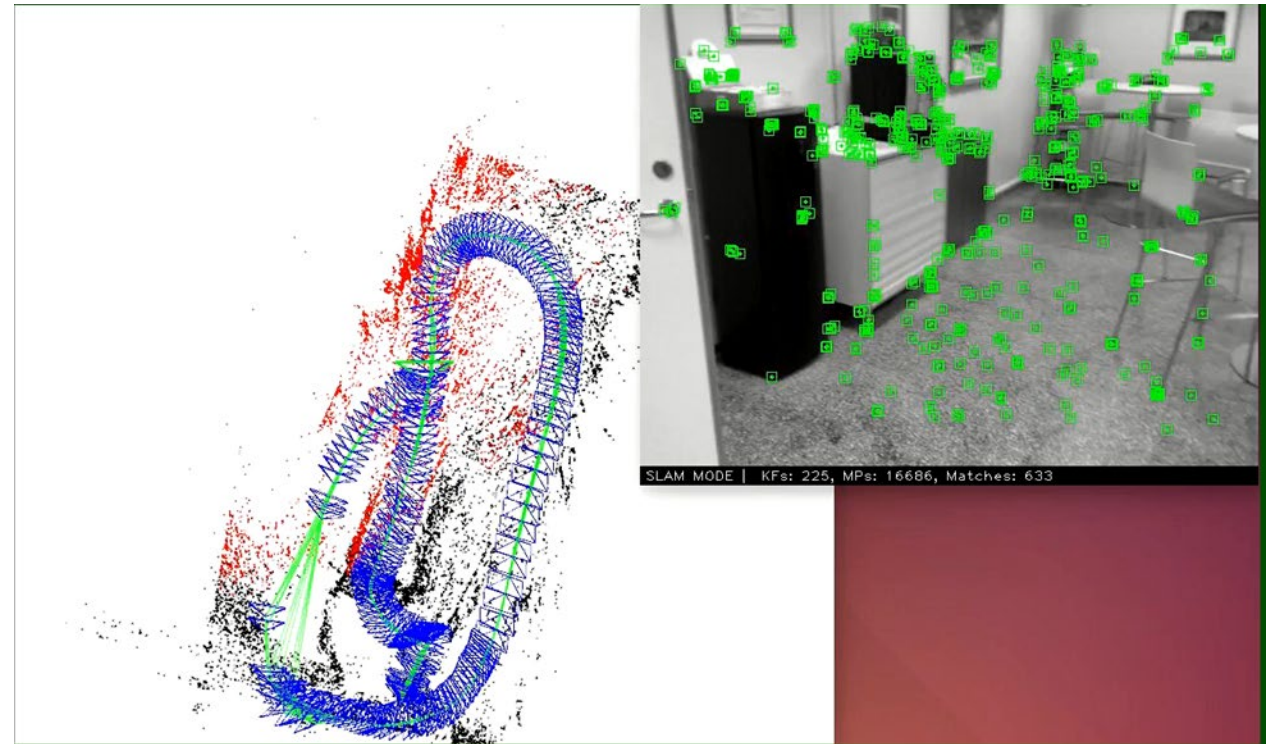
We track the map for localisation

- Estimate the camera pose relative to the map for each frame

and for building a consistent map

- Detect loop closures

These tasks have different *requirements, challenges and opportunities*



Short-term tracking for pose estimation

Requires:

- High tracking rate
- Precise pose estimate

Challenges:

- Fast correspondence search
- Many correspondences

Opportunities:

- A simple motion model often results in a good prediction for the next pose
- Conditions are almost the same, few changes
- It is often possible to significantly restrict the search for correspondences



Mid-term tracking for loop closure detection

Requires:

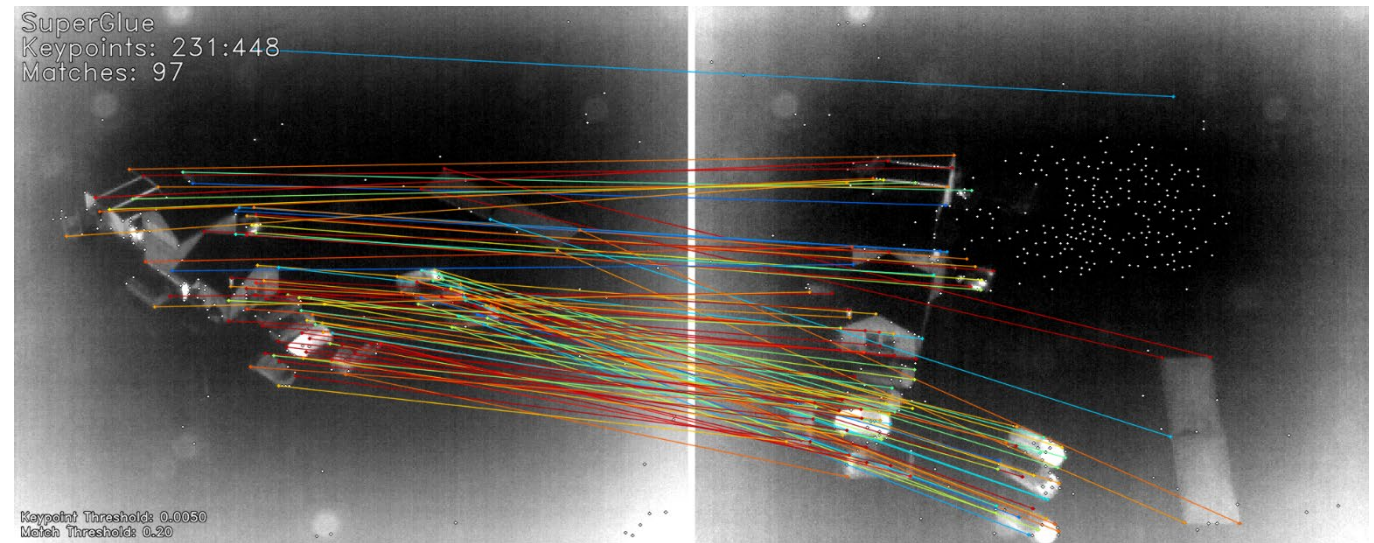
- Tracks across many views after a significant motion
- Relatively high tracking rate (keyframe rate)

Challenges:

- Different viewpoints
- Occlusions
- Several candidate keyframes

Opportunities:

- Do not need to run in frame rate
- We are close to previous keyframes
- We can restrict our search and exploit longer processing time



<https://github.com/magicLeap/SuperGluePretrainedNetwork>
Sarlin, P. E., Detone, D., Malisiewicz, T., & Rabinovich, A. (2020). SuperGlue: Learning Feature Matching with Graph Neural Networks. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 4937–4946.

Long-term tracking for loop closure detection

Requires:

- Tracks across many views after a significant time
- Global search

Additional challenges:

- Changing conditions
- Changing scene
- A very large amount of candidate keyframes

Opportunities:

- We can exploit even longer processing time



a) weather



b) season



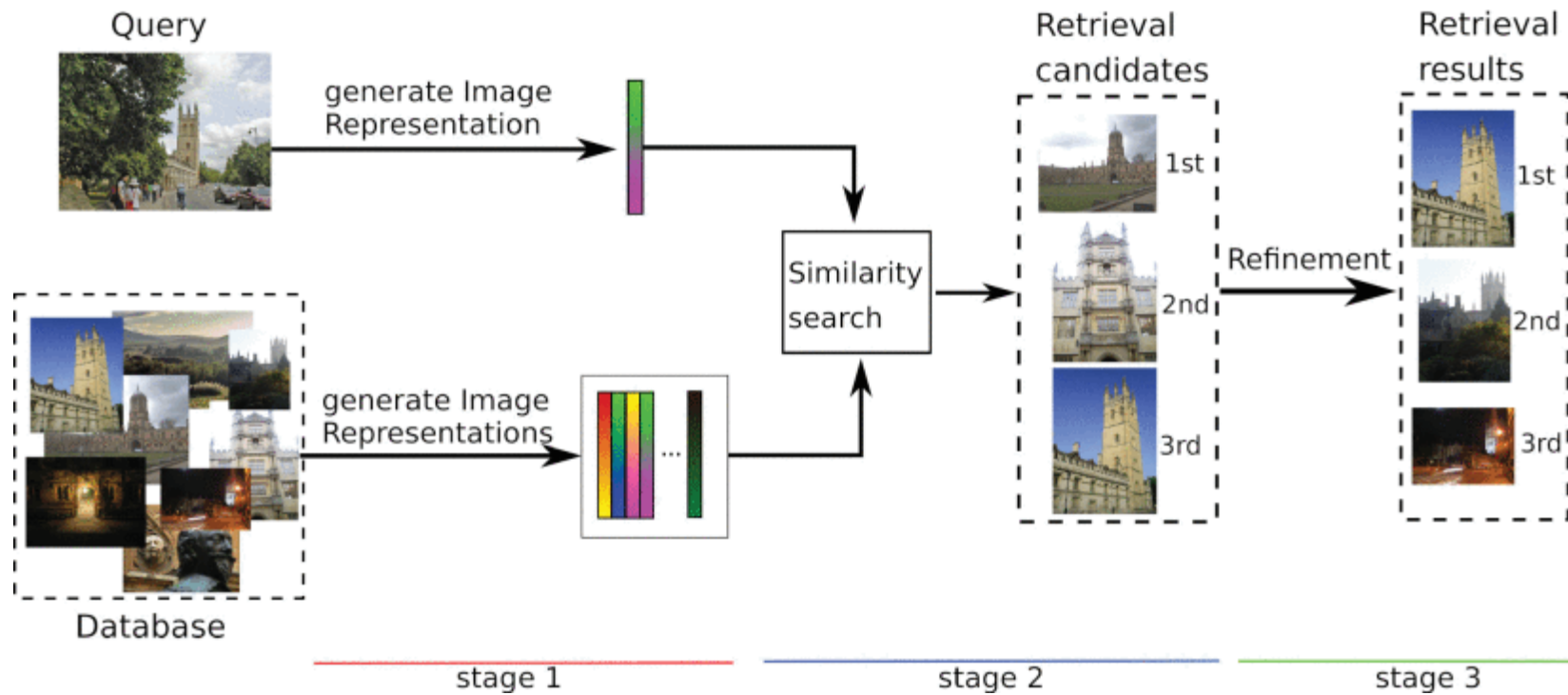
c) occlusions



d) day/night

"A Survey on Deep Visual Place Recognition," C. Masone and B. Caputo, IEEE Access, vol. 9, pp. 19516-19547, 2021

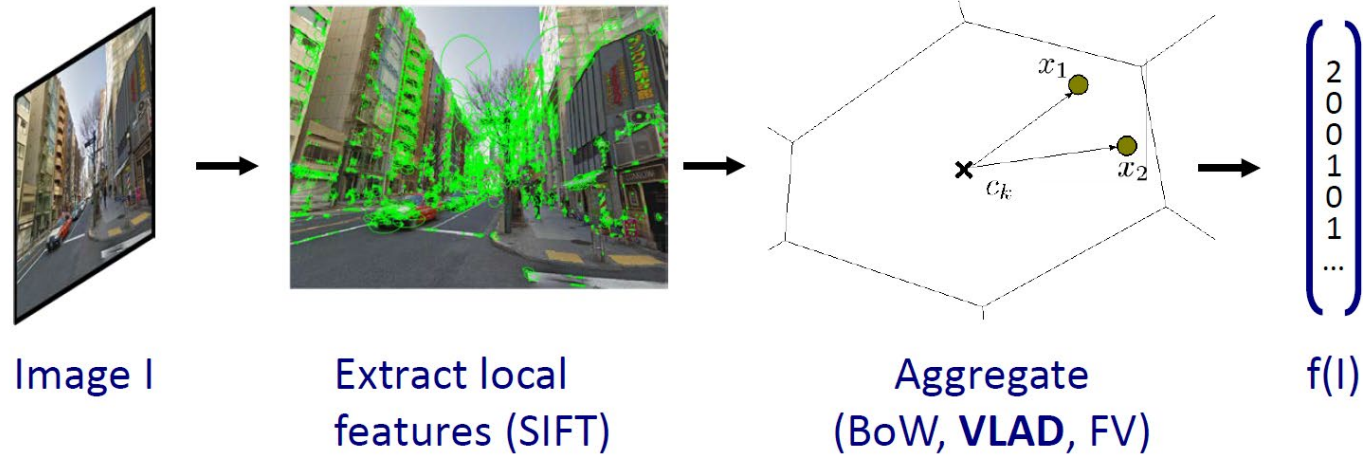
Image retrieval



"A Survey on Deep Visual Place Recognition," C. Masone and B. Caputo, IEEE Access, vol. 9, pp. 19516-19547, 2021

Image retrieval architectures

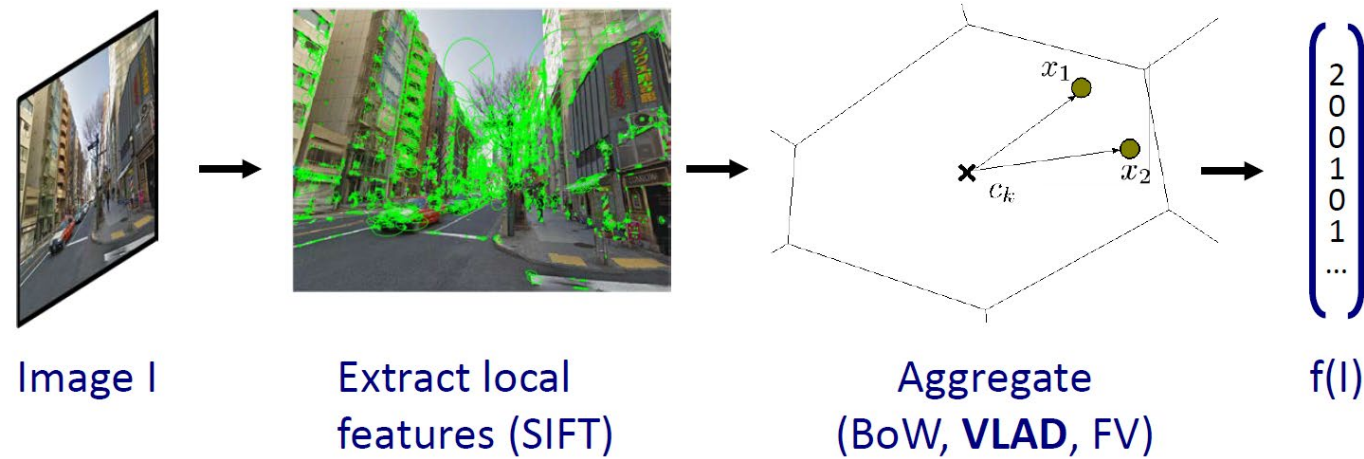
Classical
approach



"Cross-weather-time, long term Visual Geo-Localization", R. Kumar, CVPR 2021 tutorial on Cross-view and Cross-modal Visual GeoLocalization

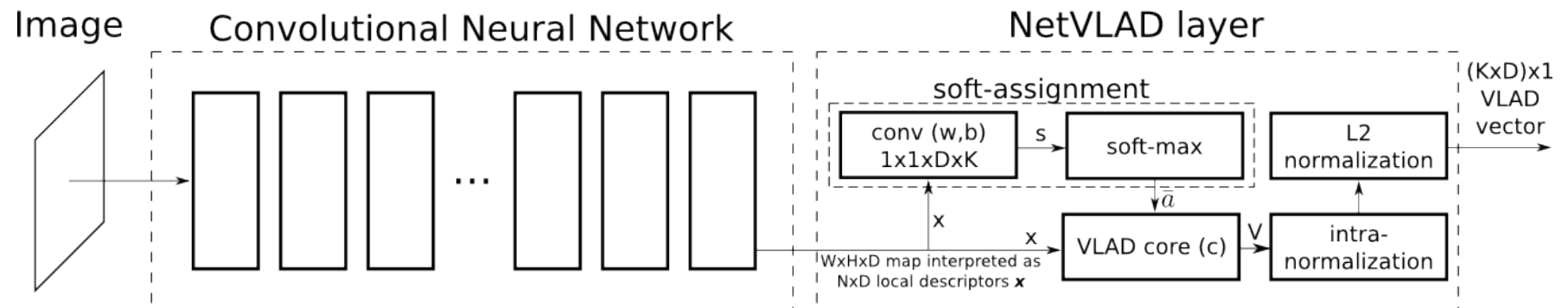
Image retrieval architectures

Classical approach



"Cross-weather-time, long term Visual Geo-Localization", R. Kumar, CVPR 2021 tutorial on Cross-view and Cross-modal Visual GeoLocalization

Trained end-to-end



Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., & Sivic, J. (2018). NetVLAD: CNN Architecture for Weakly Supervised Place Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6), 1437–1451
<https://www.di.ens.fr/willow/research/netvlad/>

Supplementary material

“Visual Place Recognition: A Survey“,
Lowry, S. et al., IEEE Transactions on Robotics, 32 (1), pp 1–19, 2016
<https://ieeexplore.ieee.org/document/7339473>

"A Survey on Deep Visual Place Recognition,"
C. Masone and B. Caputo, IEEE Access, vol. 9, pp. 19516-19547, 2021
doi: 10.1109/ACCESS.2021.3054937.

“Cross-weather-time, long term Visual Geo-Localization”
R. Kumar, CVPR 2021 tutorial on Cross-view and Cross-modal Visual GeoLocalization
<https://www.sri.com/computer-vision/cvpr-2021-tutorial-on-cross-view-and-cross-modal-visual-geo-localization/>

Part III

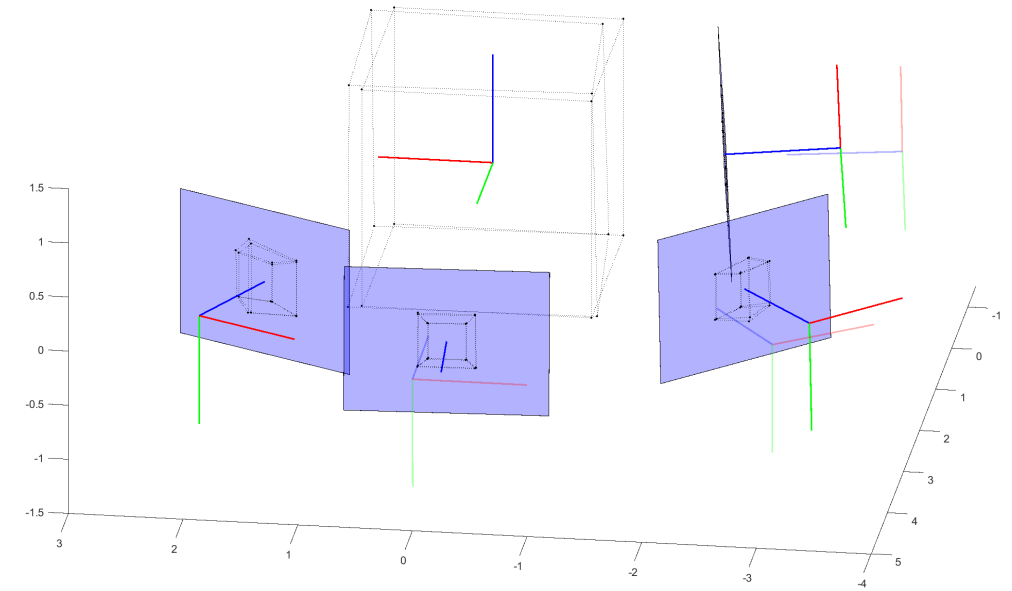
MAPPING WITH FACTOR GRAPHS

Maximum a posteriori inference

Interested in the unknown state variables X ,
given the measurements Z .

The most often used estimator for X
is the MAP estimate:

$$\begin{aligned} X^{\text{MAP}} &= \operatorname{argmax}_X p(X | Z) \\ &= \operatorname{argmax}_X \frac{p(Z | X)p(X)}{p(Z)} \\ &= \operatorname{argmax}_X l(X; Z)p(X) \\ l(X; Z) &\propto p(Z | X) \end{aligned}$$



Maximum a posteriori inference

Measurement model:

$$\mathbf{z}_i = h_i(X_i) + \eta, \quad \eta \sim N(\mathbf{0}, \Sigma_i)$$

Measurement prediction function:

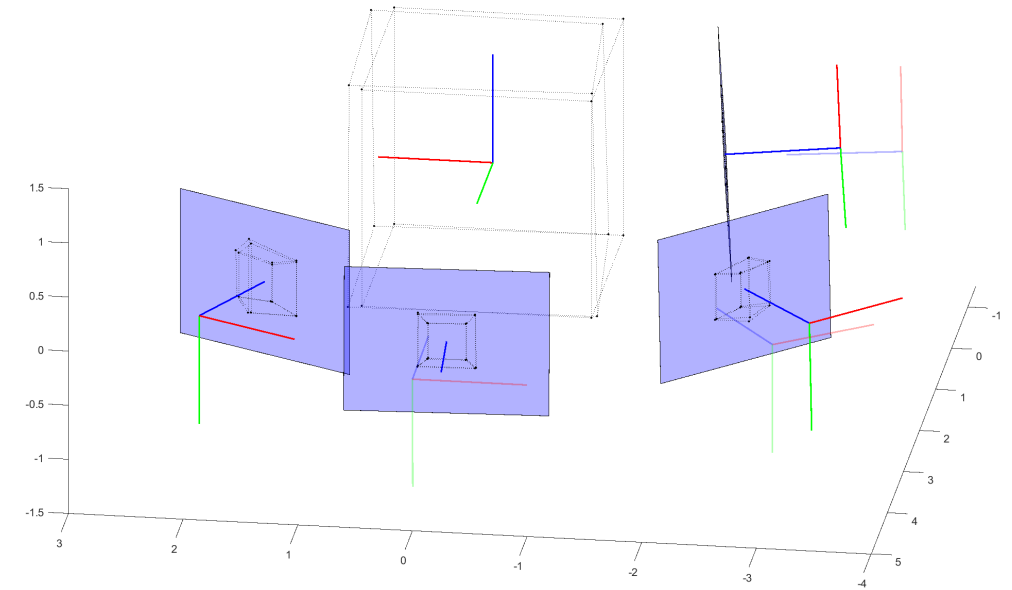
$$\hat{\mathbf{z}}_i = h_i(X_i)$$

Measurement likelihood:

$$p(\mathbf{z}_i | X_i) \propto l(X_i; \mathbf{z}_i) = \exp\left(-\frac{1}{2} \|h_i(X_i) - \mathbf{z}_i\|_{\Sigma_i}^2\right)$$

MAP estimate:

$$X^{\text{MAP}} = \operatorname{argmin}_X \sum_i \|h_i(X_i) - \mathbf{z}_i\|_{\Sigma_i}^2$$



$$\{\mathbf{T}_{wc_i}^*, \mathbf{x}_j^{w*}\} = \operatorname{argmin}_{\mathbf{T}_{wc_i}, \mathbf{x}_j^w} \sum_i \sum_j \left\| \pi_i(\mathbf{T}_{wc_i}^{-1} \cdot \mathbf{x}_j^w) - \mathbf{u}_j^i \right\|^2$$

Maximum a posteriori inference

Measurement model:

$$\mathbf{z}_i = h_i(X_i) + \eta, \quad \eta \sim N(\mathbf{0}, \Sigma_i)$$

Measurement prediction function:

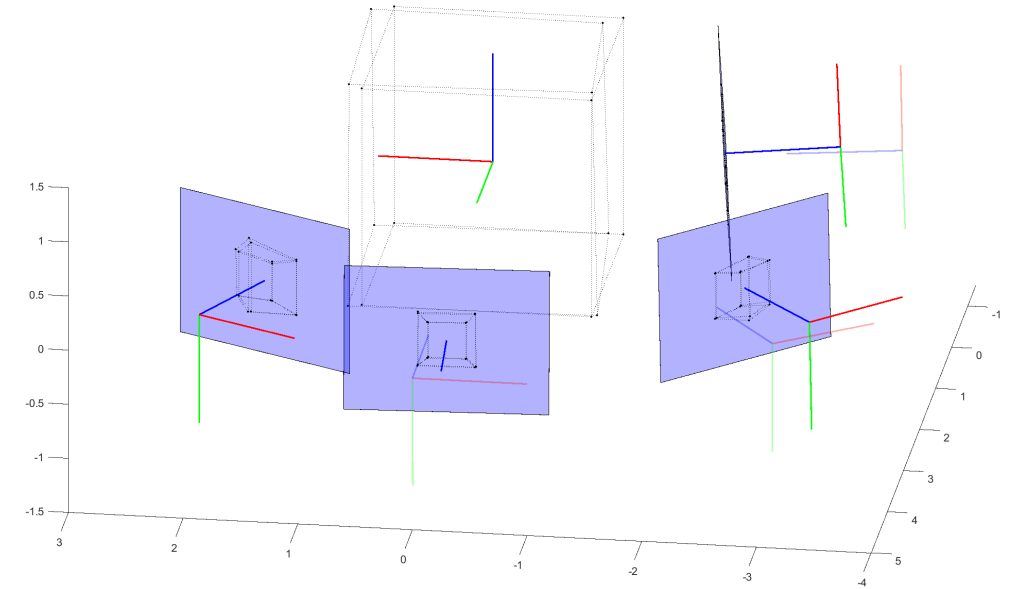
$$\hat{\mathbf{z}}_i = h_i(X_i)$$

Measurement likelihood:

$$p(\mathbf{z}_i | X_i) \propto l(X_i; \mathbf{z}_i) = \exp\left(-\frac{1}{2} \|h_i(X_i) - \mathbf{z}_i\|_{\Sigma_i}^2\right)$$

MAP estimate:

$$X^{\text{MAP}} = \underset{X}{\operatorname{argmin}} \sum_i \|h_i(X_i) - \mathbf{z}_i\|_{\Sigma_i}^2$$



Applying the MAP framework

This results in the linearised weighted least squares problem

$$\begin{aligned} \boldsymbol{\tau}^* &= \underset{\boldsymbol{\tau}}{\operatorname{argmin}} \sum_{i=1}^k \sum_{j=1}^n \| \mathbf{P}_{ij} \boldsymbol{\xi}_i + \mathbf{S}_{ij} \delta \mathbf{x}_j - \mathbf{b}_{ij} \|^2 \\ &= \underset{\boldsymbol{\tau}}{\operatorname{argmin}} \| \mathbf{A} \boldsymbol{\tau} - \mathbf{b} \|^2, \end{aligned}$$

where

$$\begin{aligned} \mathbf{P}_{ij} &= \Sigma_{n\ ij}^{-1/2} \mathbf{J}_{\mathbf{T}_{wc_i}}^{h_{ij}} \\ \mathbf{S}_{ij} &= \Sigma_{n\ ij}^{-1/2} \mathbf{J}_{\mathbf{x}_j^w}^{h_{ij}} \\ \mathbf{b}_{ij} &= \Sigma_{n\ ij}^{-1/2} (\mathbf{x}_{n\ j}^i - h_{ij}(\mathbf{T}_{wc_i}, \mathbf{x}_j^w)), \end{aligned}$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{P}_{11} & & \mathbf{S}_{11} & & \\ \vdots & & & \ddots & \\ \mathbf{P}_{1n} & & & & \mathbf{S}_{1n} \\ & \ddots & & & \\ & & \mathbf{P}_{k1} & \mathbf{S}_{k1} & \\ & & \vdots & & \ddots \\ & & \mathbf{P}_{kn} & & \mathbf{S}_{kn} \end{bmatrix} \quad \boldsymbol{\tau} = \begin{bmatrix} \boldsymbol{\xi}_1 \\ \vdots \\ \boldsymbol{\xi}_k \\ \delta \mathbf{x}_1 \\ \vdots \\ \delta \mathbf{x}_n \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} \mathbf{b}_{11} \\ \vdots \\ \mathbf{b}_{1n} \\ \vdots \\ \mathbf{b}_{k1} \\ \vdots \\ \mathbf{b}_{kn} \end{bmatrix}.$$

Maximum a posteriori inference and factor graphs

Measurement model:

$$\mathbf{z}_i = h_i(X_i) + \eta, \quad \eta \sim N(\mathbf{0}, \Sigma_i)$$

Measurement prediction function:

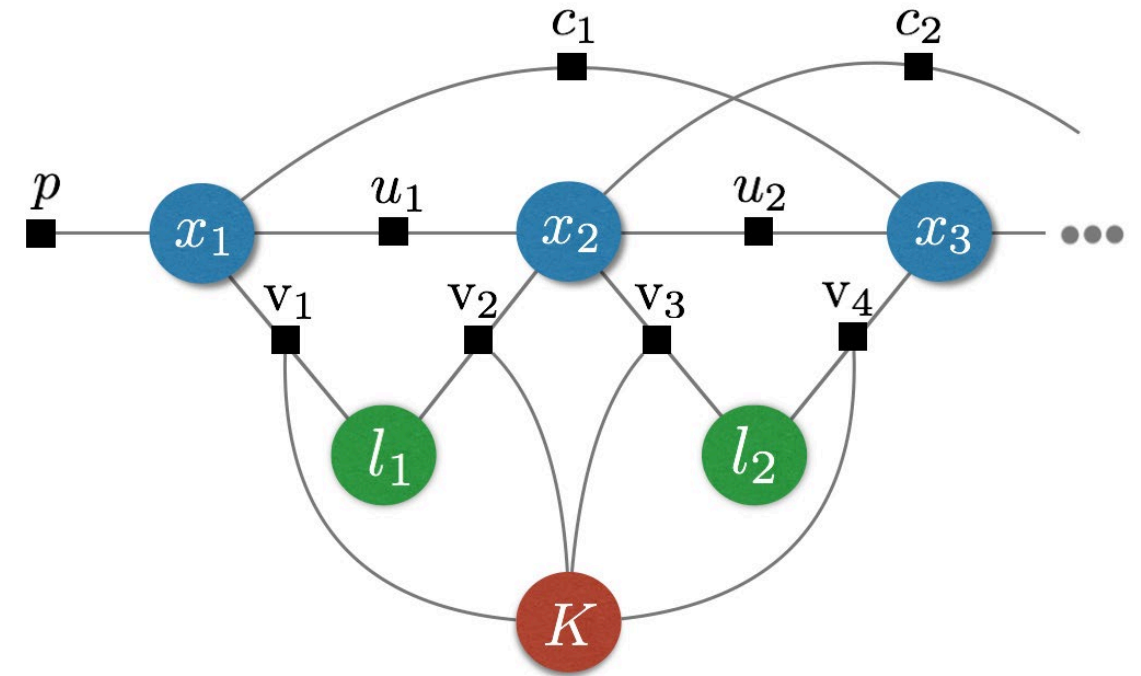
$$\hat{\mathbf{z}}_i = h_i(X_i)$$

Measurement likelihood:

$$p(\mathbf{z}_i | X_i) \propto l(X_i; \mathbf{z}_i) = \exp\left(-\frac{1}{2} \|h_i(X_i) - \mathbf{z}_i\|_{\Sigma_i}^2\right)$$

MAP estimate:

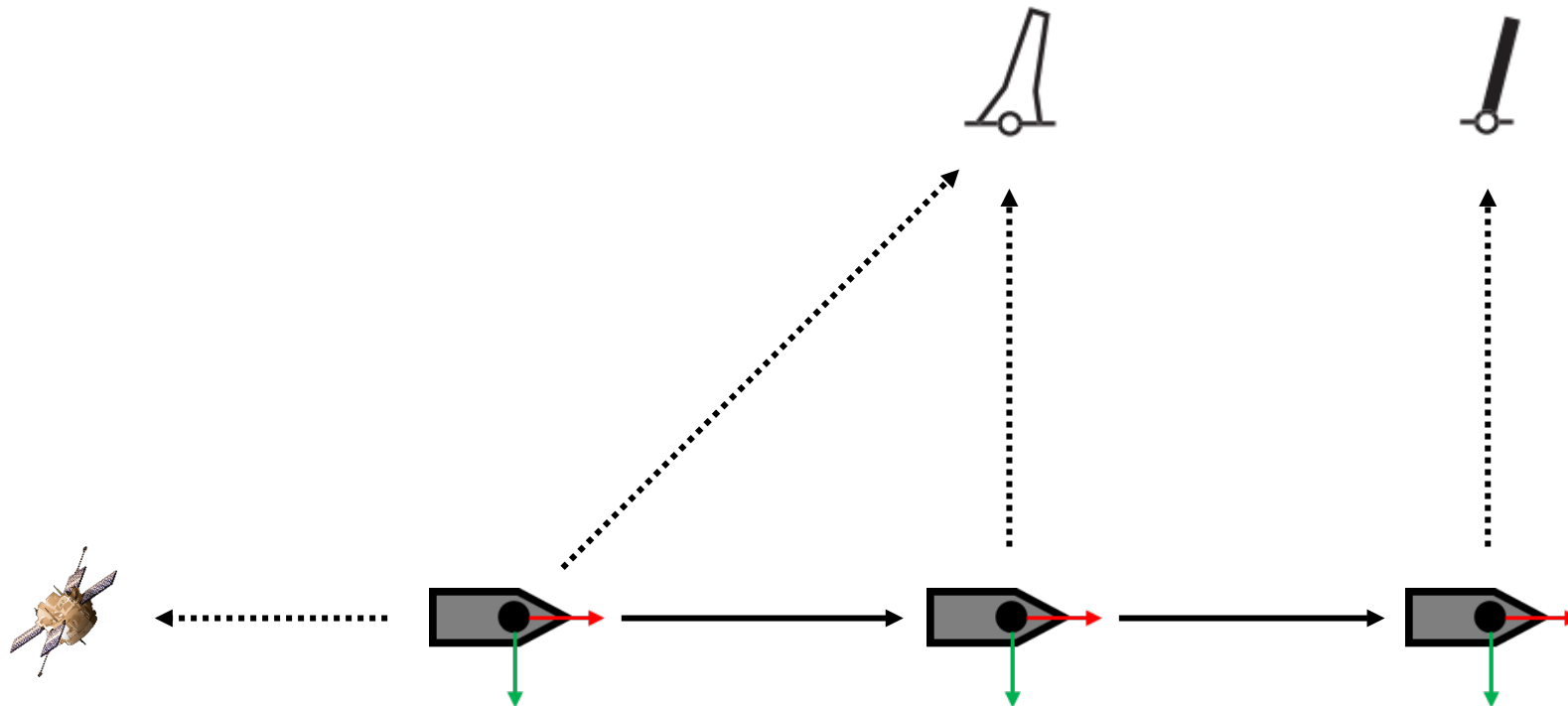
$$X^{\text{MAP}} = \underset{X}{\operatorname{argmin}} \sum_i \|h_i(X_i) - \mathbf{z}_i\|_{\Sigma_i}^2$$



Cadena, C., et al. (2016). Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age. *IEEE Transactions on Robotics*, 32(6), 1309–1332

Maximum a posteriori inference and factor graphs

Simple SLAM example



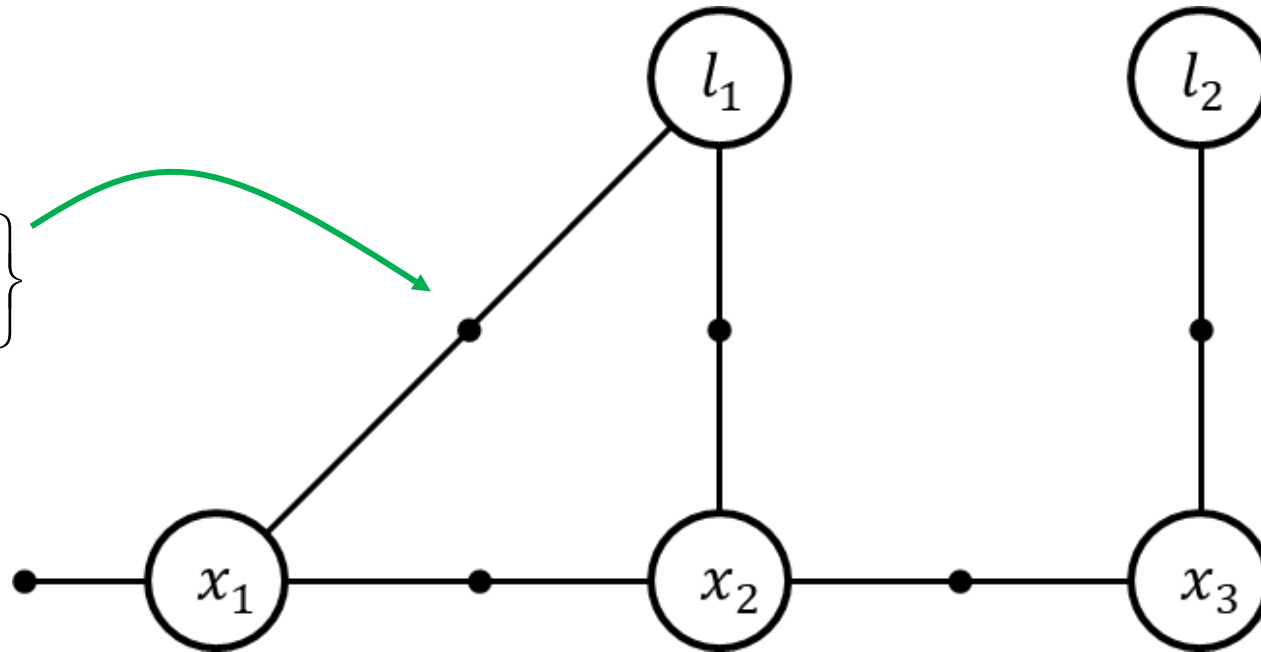
<https://github.com/tussedrotten/simple-factorgraph-example>

Maximum a posteriori inference and factor graphs

Simple SLAM example

$$\begin{bmatrix} r \\ \alpha \end{bmatrix} = \rho(\mathbf{x}^r) = \begin{bmatrix} \sqrt{x^2 + y^2} \\ \arctan\left(\frac{y}{x}\right) \end{bmatrix}$$

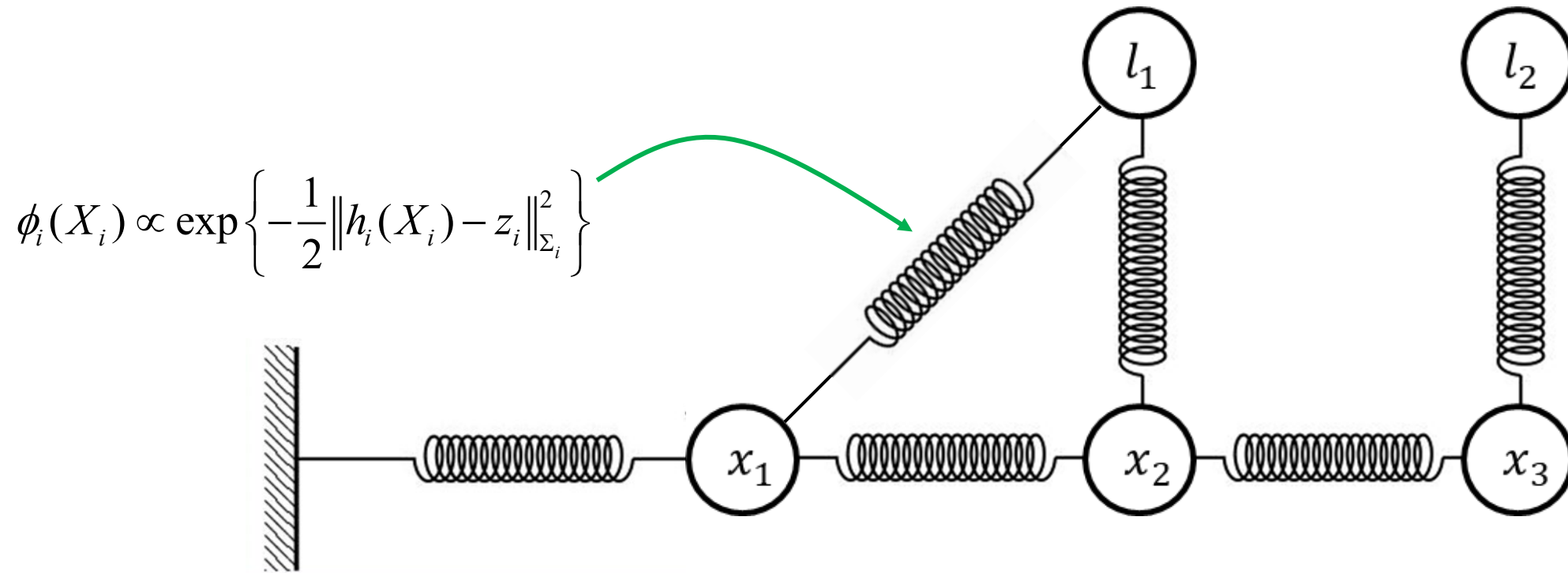
$$\phi_i(X_i) \propto \exp\left\{-\frac{1}{2}\|h_i(X_i) - z_i\|_{\Sigma_i}^2\right\}$$



<https://github.com/tussedrotten/simple-factorgraph-example>

Maximum a posteriori inference and factor graphs

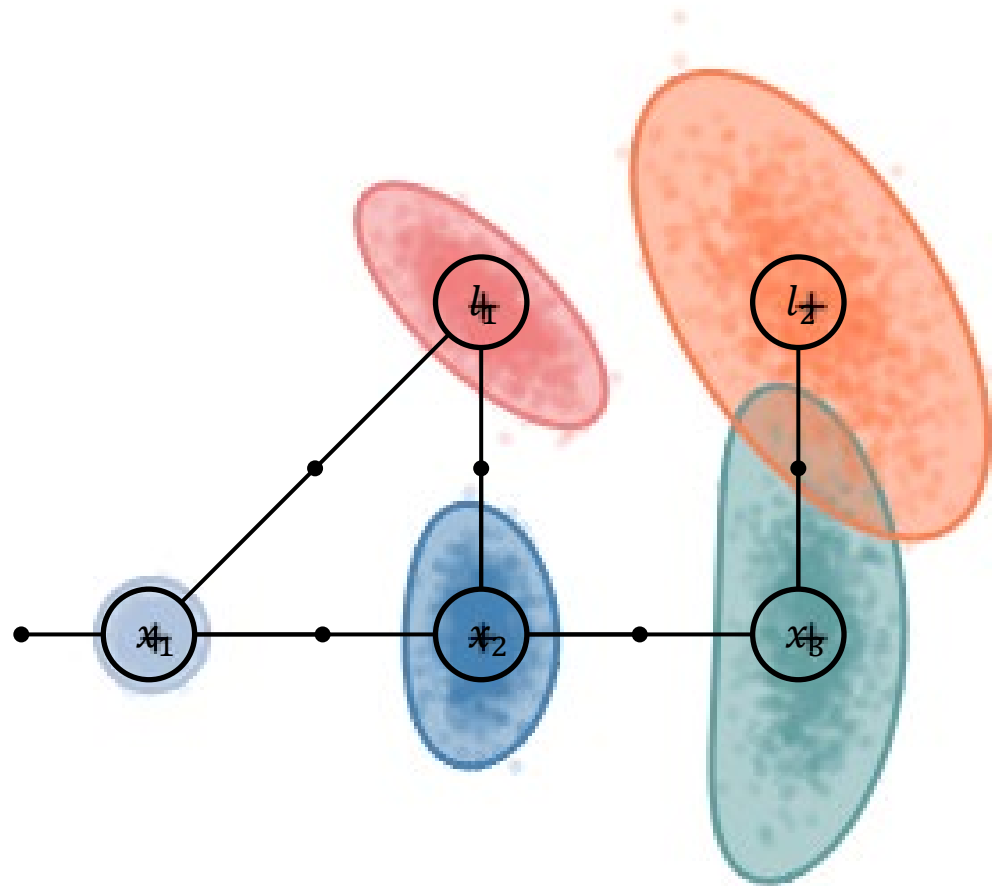
Simple SLAM example



<https://github.com/tussedrotten/simple-factorgraph-example>

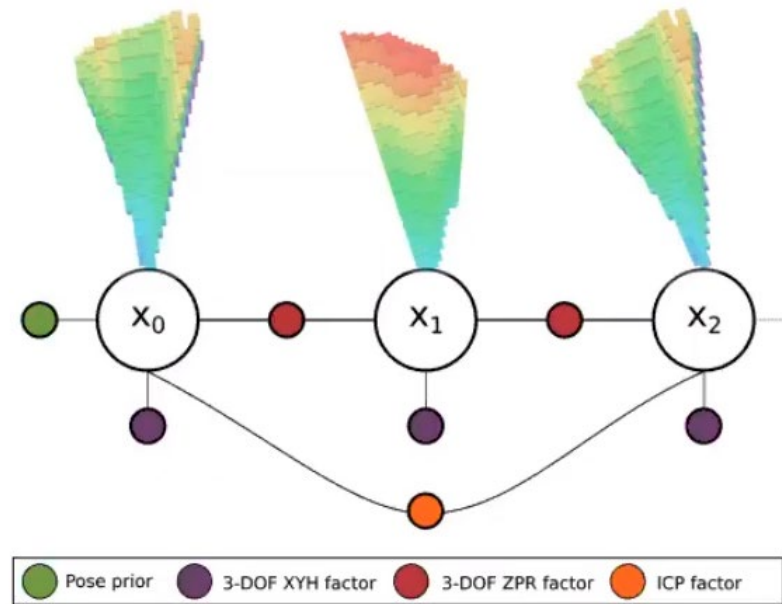
Maximum a posteriori inference and factor graphs

Simple SLAM example



<https://github.com/tussedrotten/simple-factorgraph-example>

Factor graphs make it easier to talk and think about state estimation!



$$\mathcal{X}^* = \underset{\mathcal{X}}{\operatorname{argmin}} \left(\sum_{i=1}^N \underbrace{(\|\mathcal{U}(x_{i-1}, x_i) - u_i\|_{\Psi_i}^2)}_{\text{XYH factor}} + \underbrace{\|\mathcal{V}(x_i) - v_i\|_{\Phi_i}^2}_{\text{ZPR factor}} \right. \\ \left. + \sum_{(i,k) \in \text{LC}} \underbrace{(\|\mathcal{L}(x_i, x_k) - l_{ik}\|_{\Gamma_{i,k}}^2)}_{\text{loop closure factor}} + \underbrace{\|\mathbf{p}_0 \ominus \mathbf{x}_0\|_{\Sigma_0}^2}_{\text{prior factor}} \right)$$

S. Suresh, P. Sodhi, J. G. Mangelson, D. Wettergreen and M. Kaess, "Active SLAM using 3D Submap Saliency for Underwater Volumetric Exploration," 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 2020, pp. 3132-3138, doi: 10.1109/ICRA40945.2020.9196939.

Supplementary material

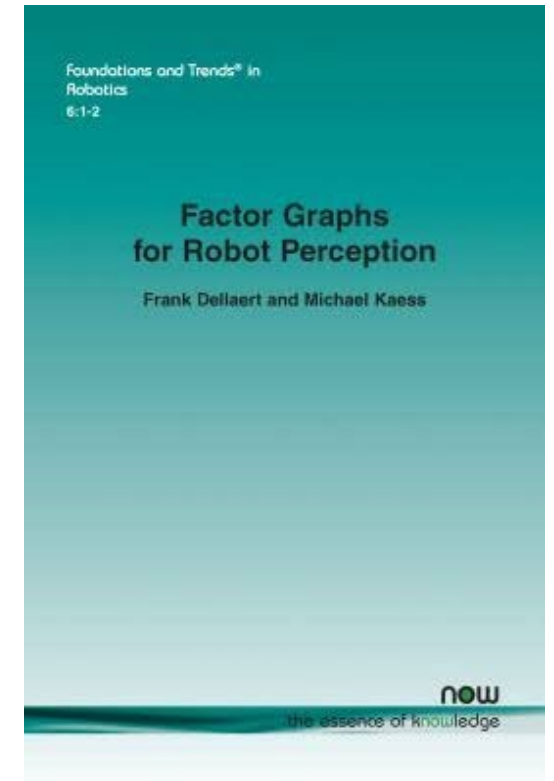
Georgia Tech Smoothing and Mapping library

- <https://gtsam.org/>
- <https://github.com/borglab/gtsam>

Tutorial: <https://gtsam.org/tutorials/intro.html>

Factor Graphs for Robot Perception
by Frank Dellaert and Michael Kaess

<https://www.cc.gatech.edu/~dellaert/pubs/Dellaert17fnt.pdf>



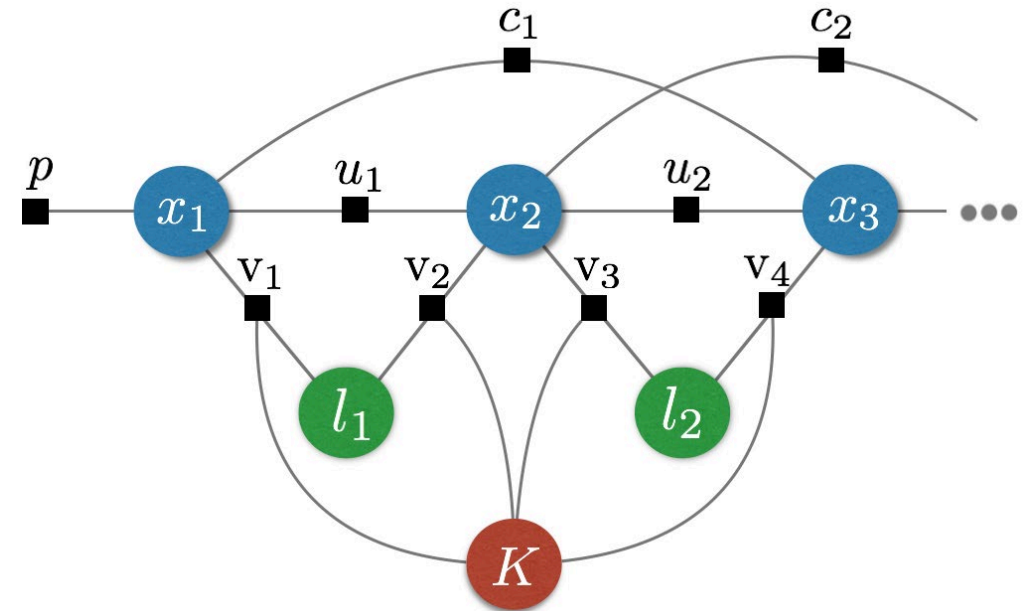
Part IV

VSLAM BACKEND STRATEGIES

Batch processing

De facto standard is to formulate the mapping problem as a **batch MAP estimation problem!**

- Generally more accurate
- Allows long-term loop-closure correction
- But the problem grows over time
→ Real-time batch inference not feasible?



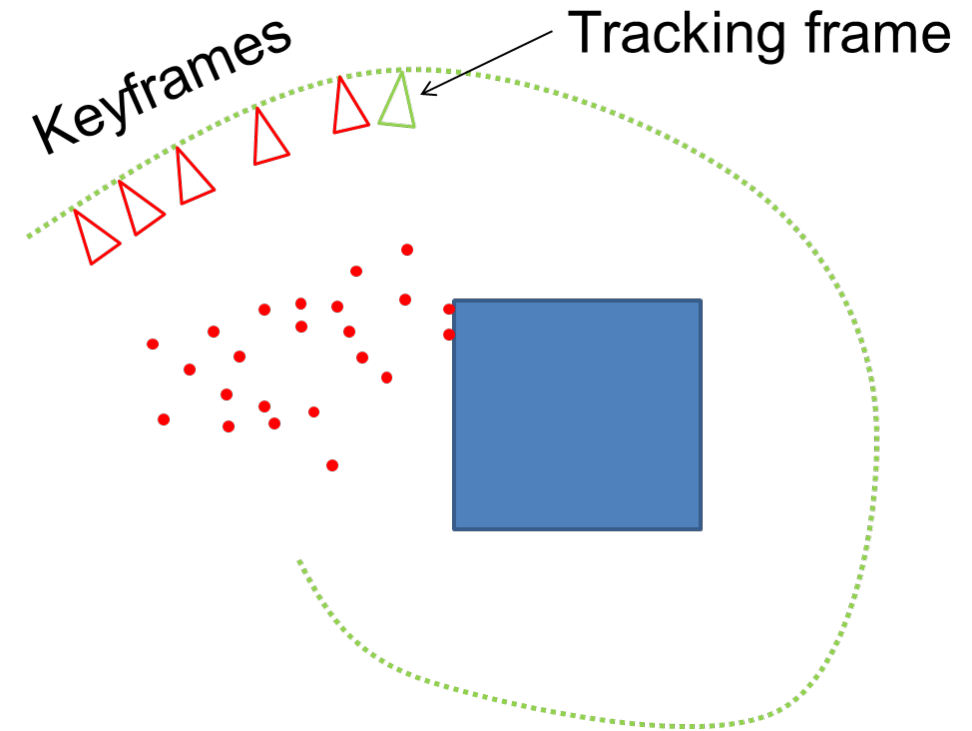
Cadena, C., et al. (2016). Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age. *IEEE Transactions on Robotics*, 32(6), 1309–1332
Strasdat, H., Montiel, J. M. M., & Davison, A. J. (2012). Visual SLAM: Why filter? *Image and Vision Computing*, 30(2), 65–77

Full bundle adjustment over keyframes

Track every frame

Map with keyframes only

Parallel tracking and mapping
with full bundle adjustment



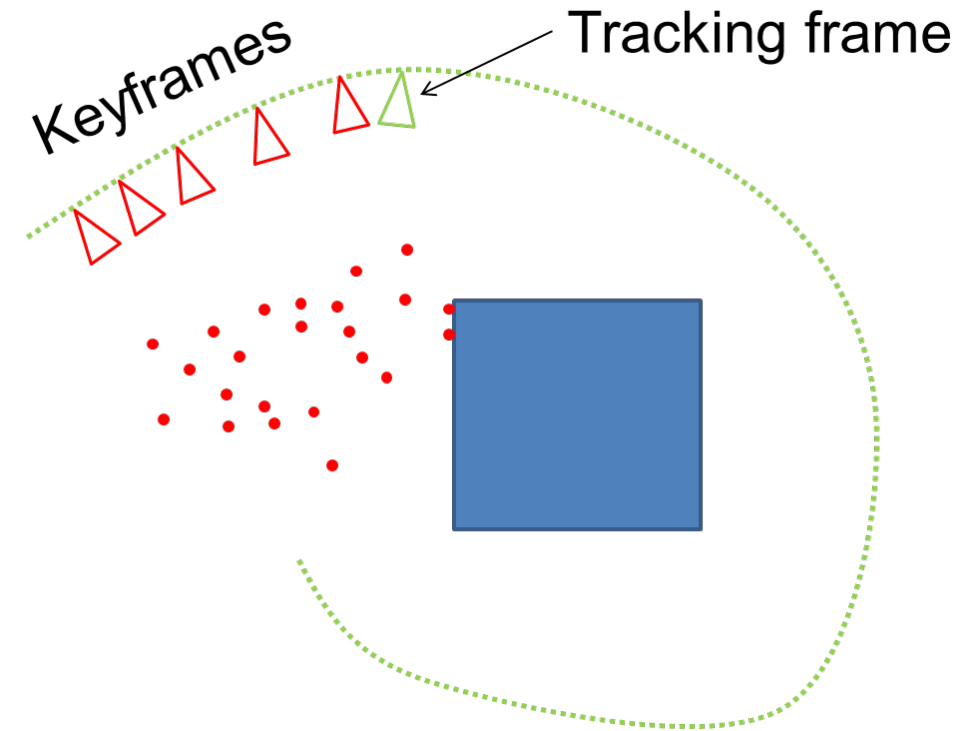
Full bundle adjustment over keyframes

Track every frame

Map with keyframes only

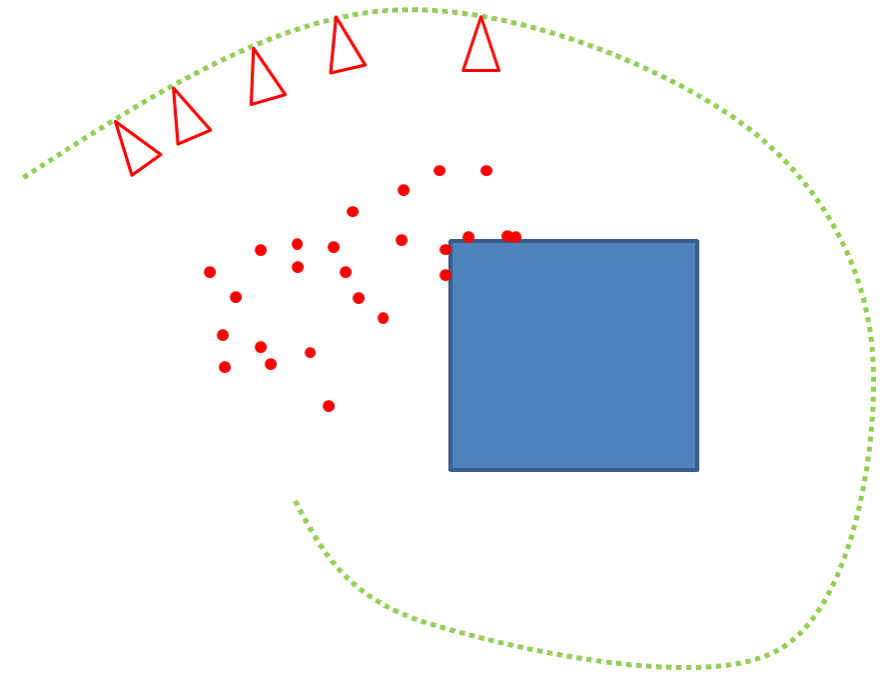
Parallel tracking and mapping
with full bundle adjustment

- Map still grows unbounded
when exploring



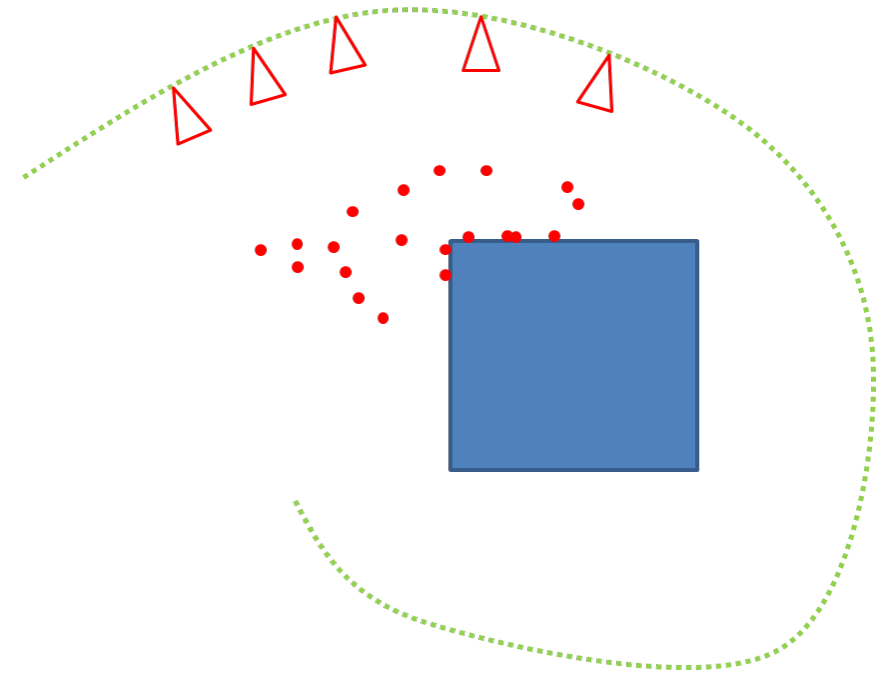
Fixed-lag bundle adjustment

Perform BA over a **fixed-lag**
of the last n keyframes



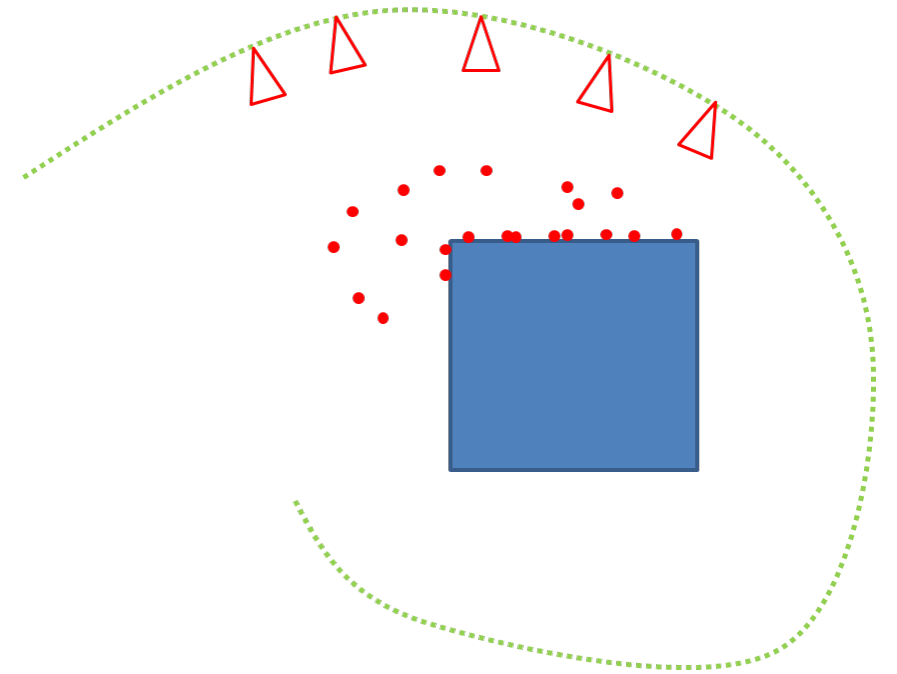
Fixed-lag bundle adjustment

Perform BA over a **fixed-lag**
of the last n keyframes



Fixed-lag bundle adjustment

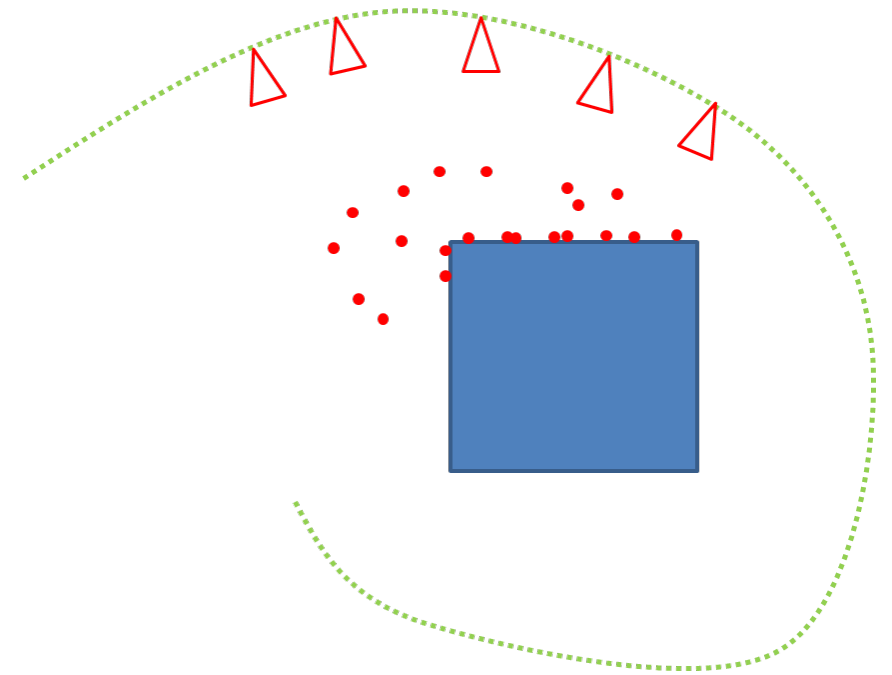
Perform BA over a **fixed-lag**
of the last n keyframes



Fixed-lag bundle adjustment

Perform BA over a **fixed-lag** of the last n keyframes

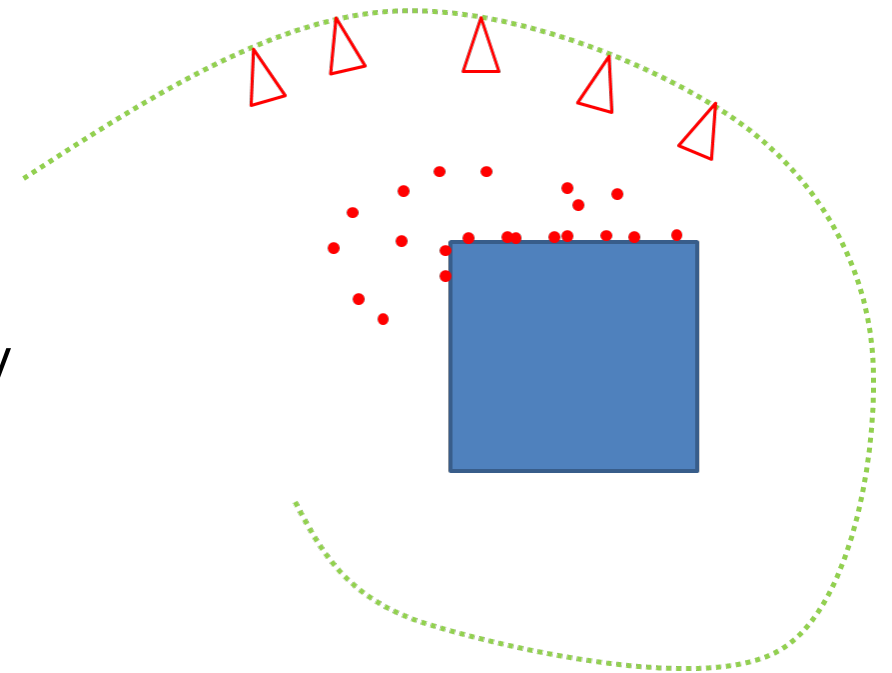
- Constant-time operation



Fixed-lag bundle adjustment

Perform BA over a **fixed-lag** of the last n keyframes

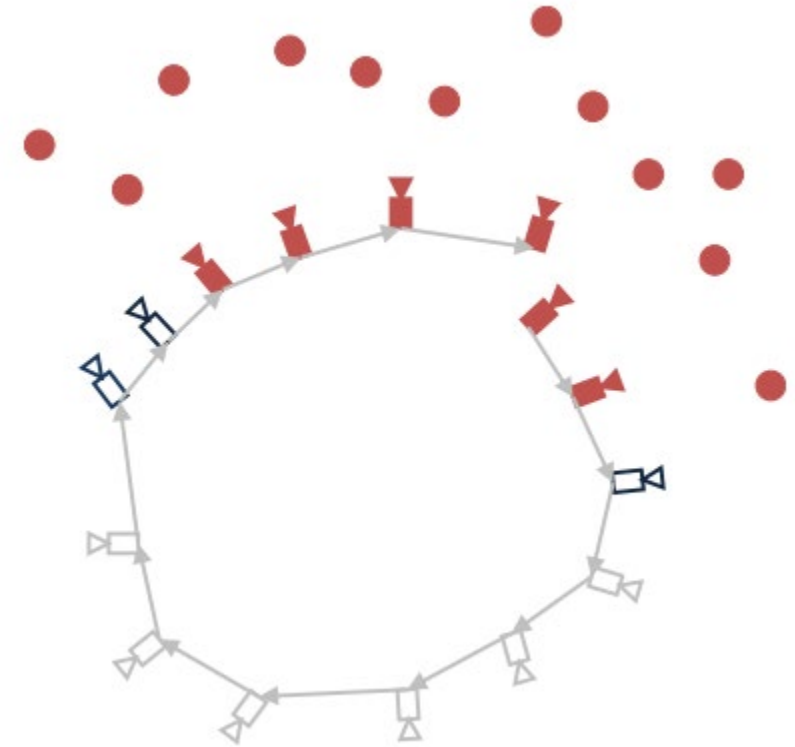
- Constant-time operation
- Marginalisation often results in dense Gaussian priors, hindering efficient inference
- Share part of the issues with filtering, such as consistency and build-up of linearisation errors
- Bounded in how far back in keyframes one may perform loop closures



Local bundle adjustment

Perform BA within an **active window** of keyframes with **co-visible points**

Keep keyframes at the boundary fixed



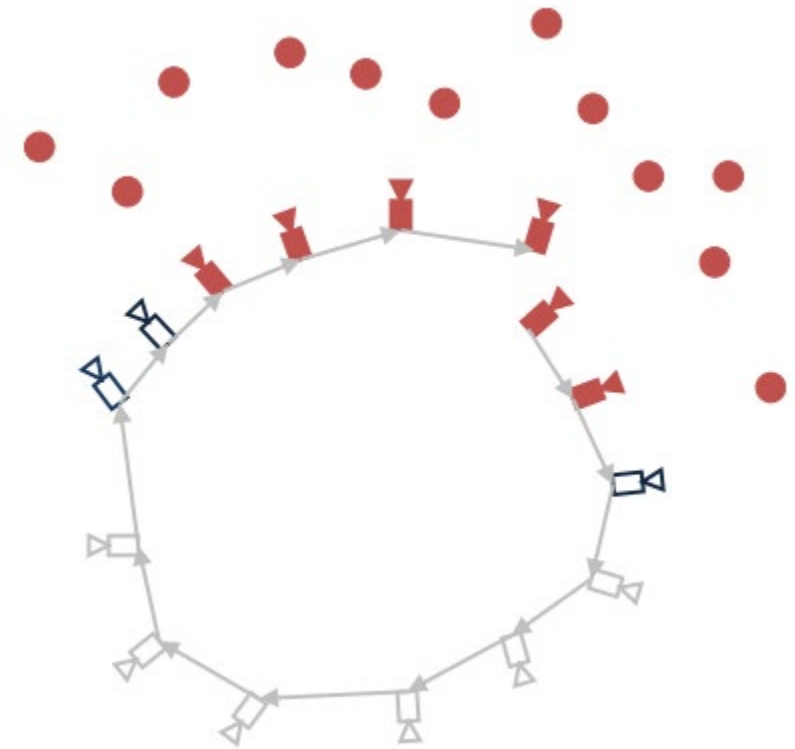
Strasdat, H., Davison, A. J., Montiel, J. M. M., & Konolige, K. (2011). Double window optimisation for constant time visual SLAM. Proceedings of the IEEE International Conference on Computer Vision, 2352–2359

Local bundle adjustment

Perform BA within an **active window** of keyframes with **co-visible points**

Keep keyframes at the boundary fixed

➤ ~Constant-time operation



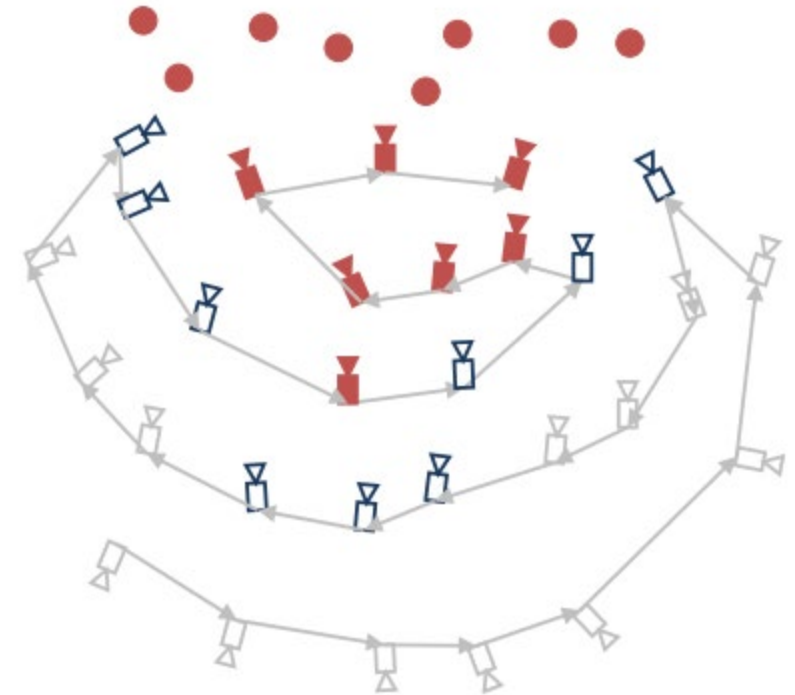
Strasdat, H., Davison, A. J., Montiel, J. M. M., & Konolige, K. (2011). Double window optimisation for constant time visual SLAM. Proceedings of the IEEE International Conference on Computer Vision, 2352–2359

Local bundle adjustment

Perform BA within an **active window** of keyframes with **co-visible points**

Keep keyframes at the boundary fixed

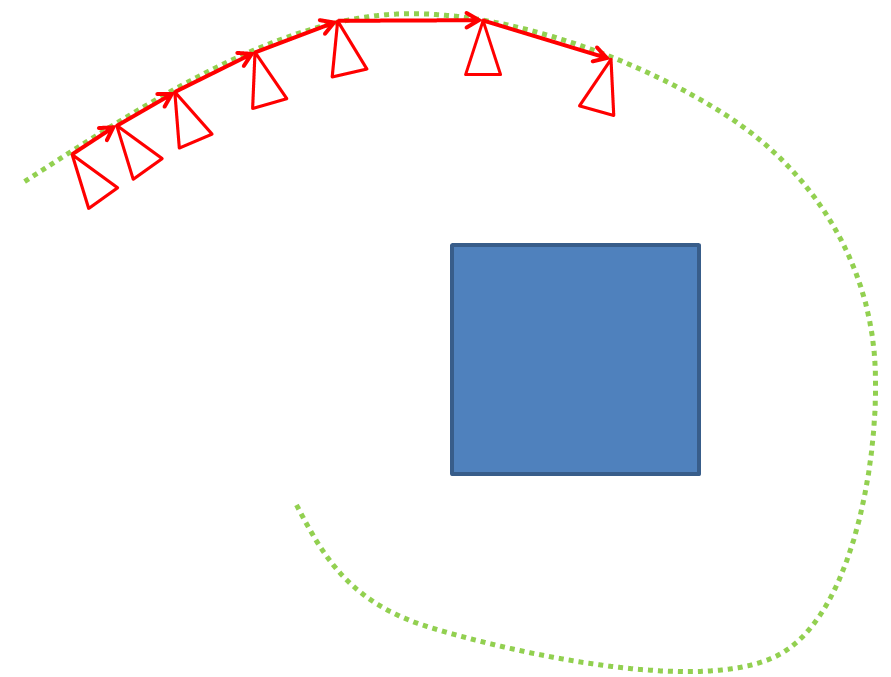
- ~Constant-time operation
- Loopy motion results in a large number of keyframes on the boundary
- Hampers convergence and accuracy



Strasdat, H., Davison, A. J., Montiel, J. M. M., & Konolige, K. (2011). Double window optimisation for constant time visual SLAM. Proceedings of the IEEE International Conference on Computer Vision, 2352–2359

Pose graph

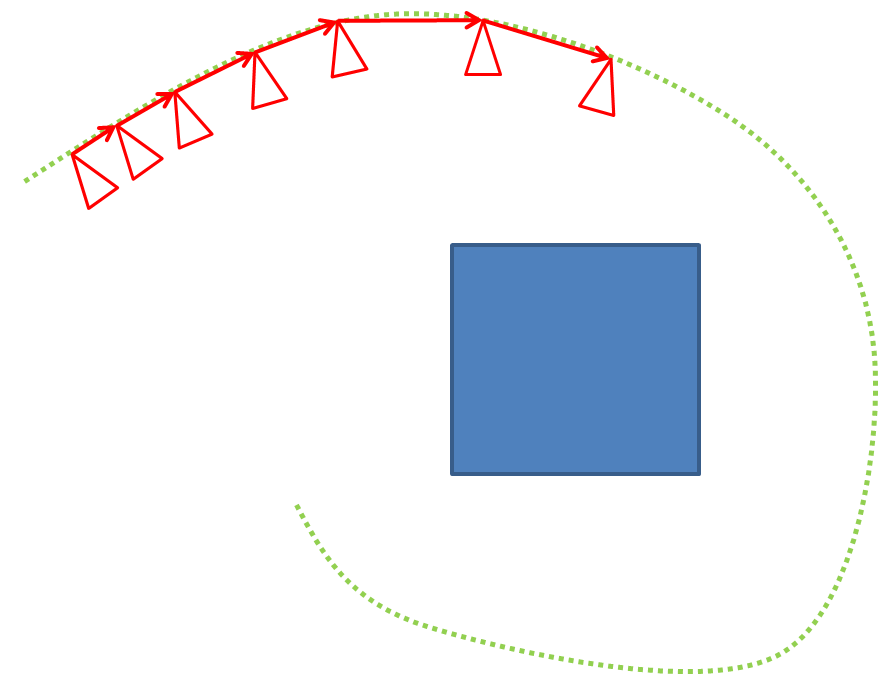
Marginalise out the points, keep only
relative pose constraints between the keyframes



Pose graph

Marginalize out the points, keep only **relative pose constraints** between the keyframes

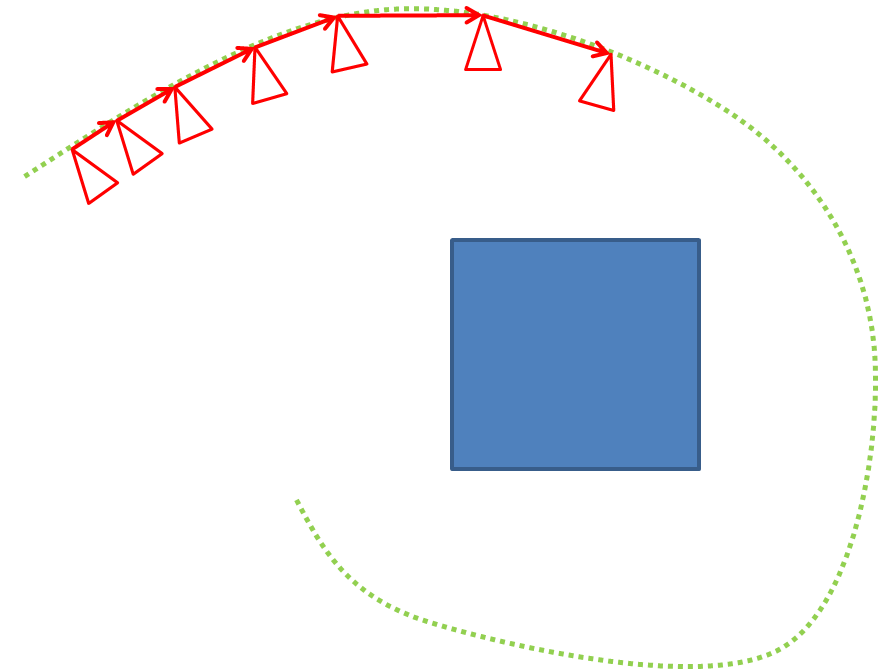
- Faster to optimise



Pose graph

Marginalize out the points, keep only **relative pose constraints** between the keyframes

- Faster to optimise
- Approximation, since these constraints do not fully encode the nonlinear connections between frames and points
- Map still grows unbounded when exploring

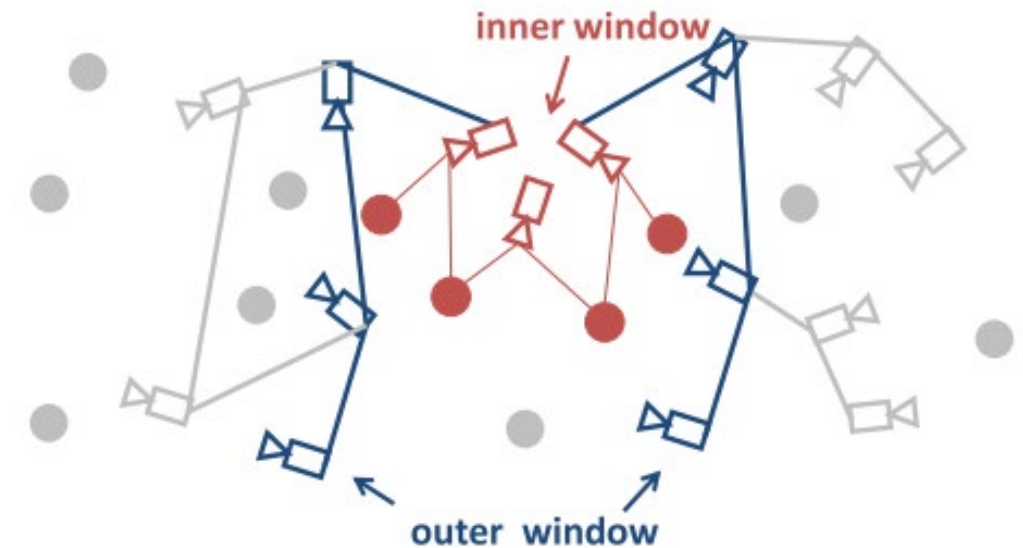


Double window optimisation

Inner window: Local bundle adjustment

Outer window: Pose graph based on co-visibility

Joint optimisation



Strasdat, H., Davison, A. J., Montiel, J. M. M., & Konolige, K. (2011). Double window optimisation for constant time visual SLAM. Proceedings of the IEEE International Conference on Computer Vision, 2352–2359

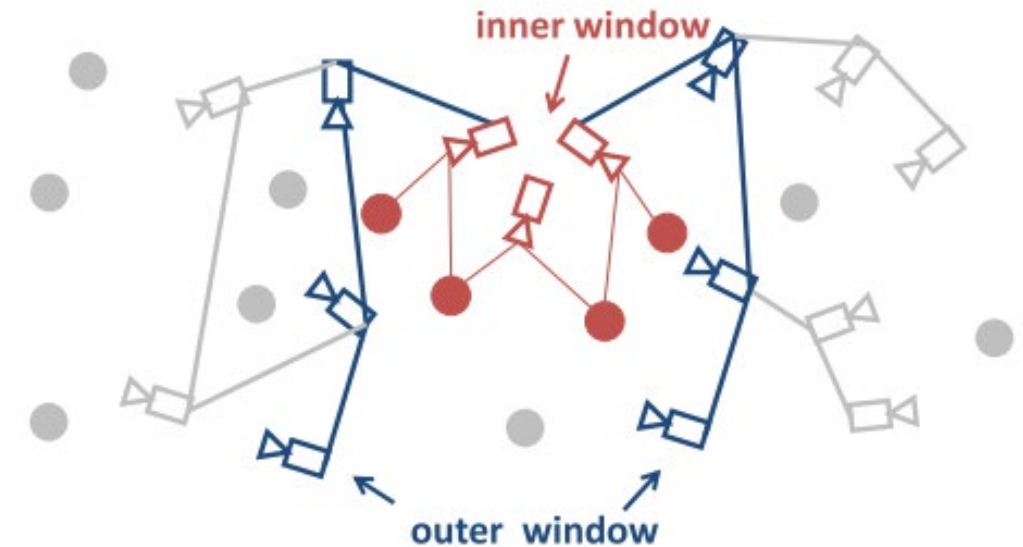
Double window optimisation

Inner window: Local bundle adjustment

Outer window: Pose graph based on co-visibility

Joint optimisation

- Locally Euclidean, globally topological
- ~Constant-time with fixed outer window



Strasdat, H., Davison, A. J., Montiel, J. M. M., & Konolige, K. (2011). Double window optimisation for constant time visual SLAM. Proceedings of the IEEE International Conference on Computer Vision, 2352–2359

Double window optimisation

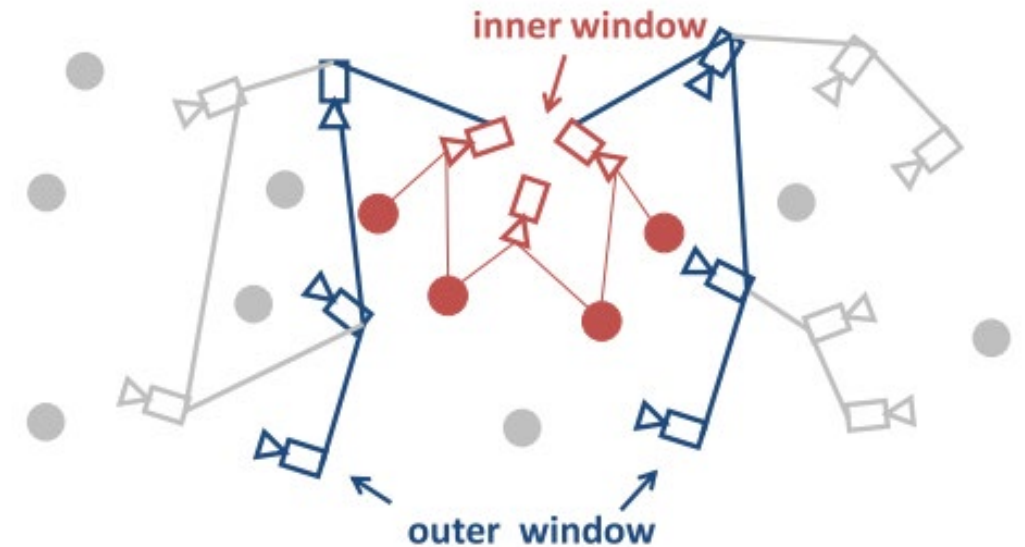
Inner window: Local bundle adjustment

Outer window: Pose graph based on co-visibility

Joint optimisation

- Locally Euclidean, globally topological
- ~Constant-time with fixed outer window

Examples: [Video1](#), [Video2](#)



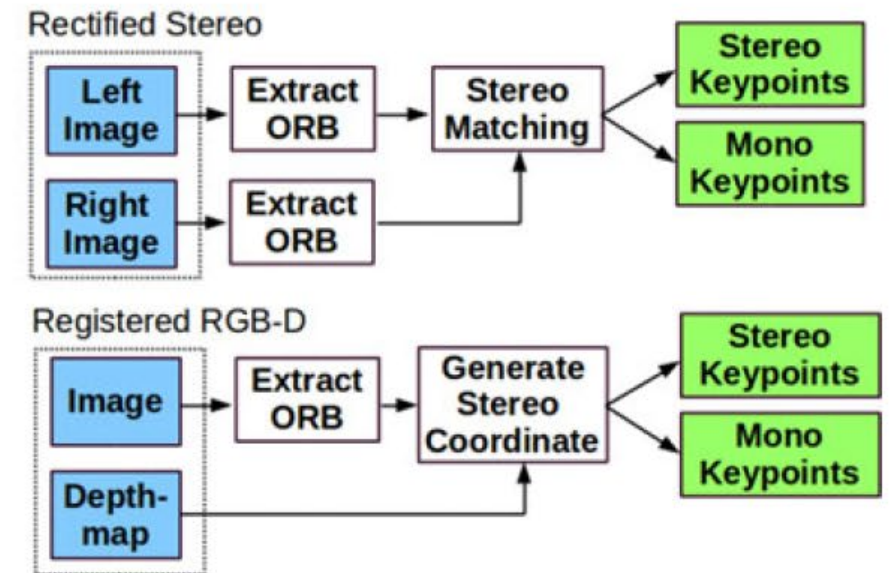
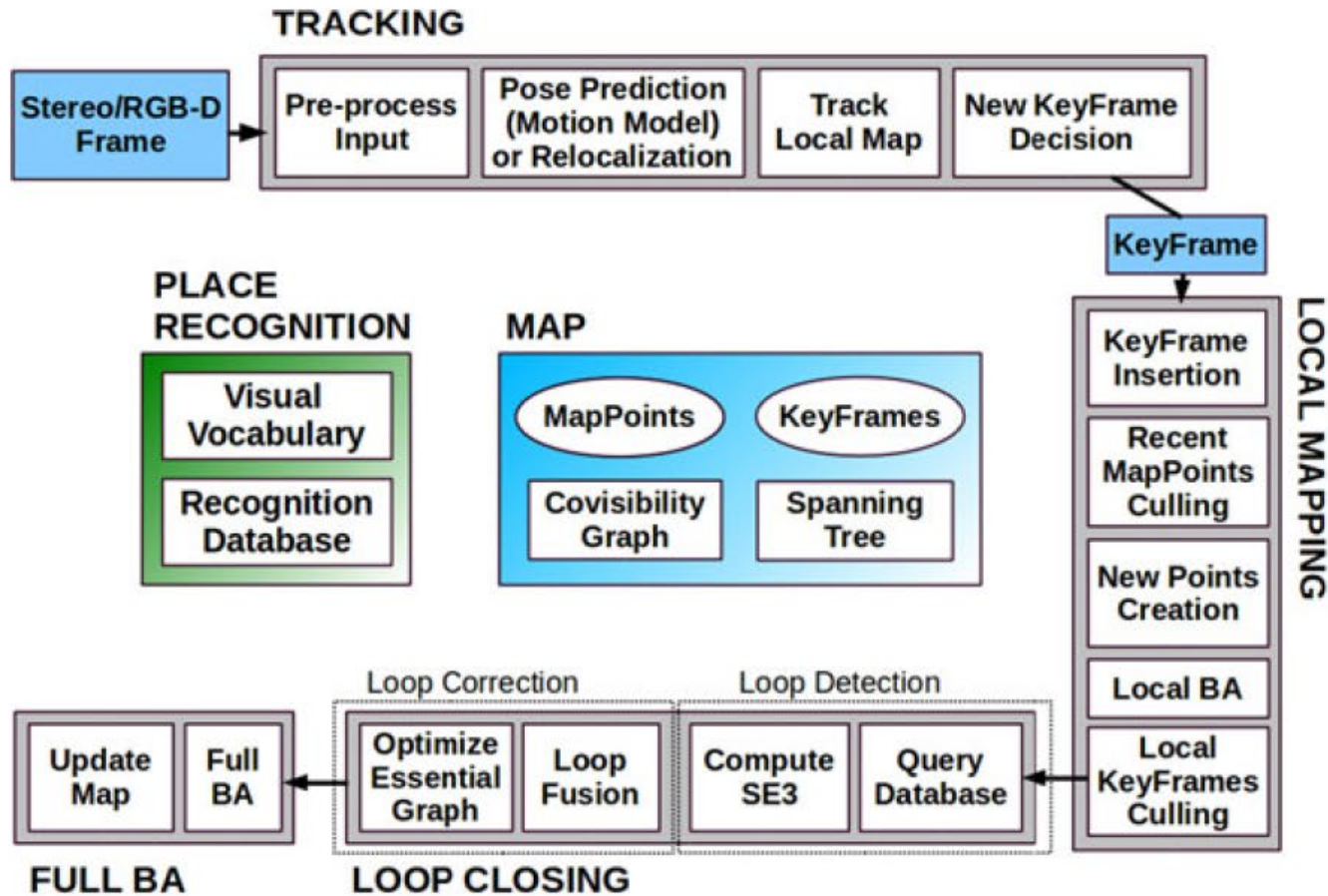
Strasdat, H., Davison, A. J., Montiel, J. M. M., & Konolige, K. (2011). Double window optimisation for constant time visual SLAM. Proceedings of the IEEE International Conference on Computer Vision, 2352–2359

Part V

VSLAM SYSTEMS

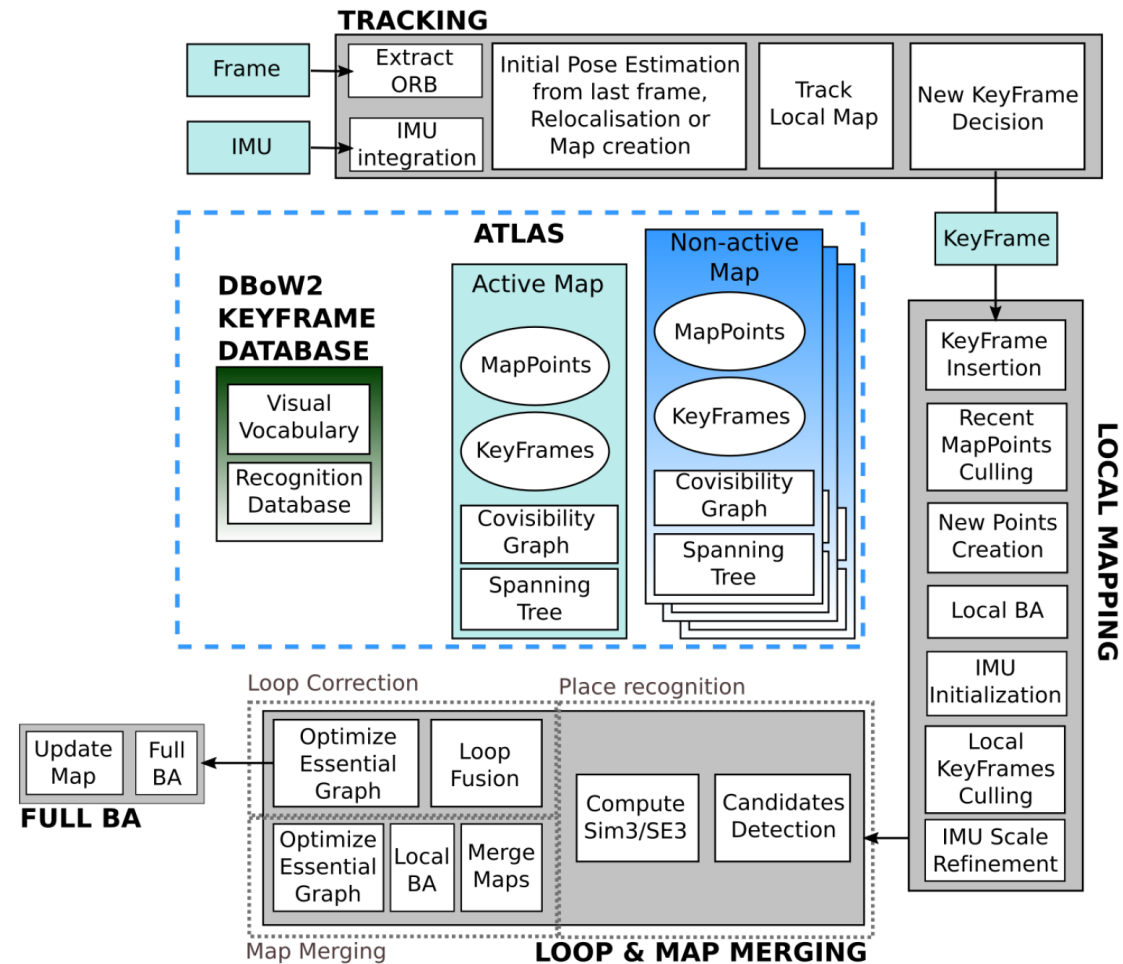
	SLAM or VO	Pixels used	Data association	Estimation	Relocali- zation	Loop closing	Multi Maps	Mono	Stereo	Mono IMU	Stereo IMU	Fisheye	Accuracy	Robustness	Open source
Mono-SLAM [13], [14]	SLAM	Shi Tomasi	Correlation	EKF	-	-	-	✓	-	-	-	-	Fair	Fair	[15] ¹
PTAM [16]–[18]	SLAM	FAST	Pyramid SSD	BA	Thumbnail	-	-	✓	-	-	-	-	Very Good	Fair	[19]
LSD-SLAM [20], [21]	SLAM	Edgelets	Direct	PG	-	FABMAP PG	-	✓	✓	-	-	-	Good	Fair	[22]
SVO [23], [24]	VO	FAST+ Hi.grad.	Direct	Local BA	-	-	-	✓	✓	-	-	✓	Very Good	Very Good	[25] ²
ORB-SLAM2 [2], [3]	SLAM	ORB	Descriptor	Local BA	DBoW2	DBoW2 PG+BA	-	✓	✓	-	-	-	Exc.	Very Good	[26]
DSO [27]–[29]	VO	High grad.	Direct	Local BA	-	-	-	✓	✓	-	-	✓	Fair	Very Good	[30]
DSM [31]	SLAM	High grad.	Direct	Local BA	-	-	-	✓	-	-	-	-	Very Good	Very Good	[32]
MSCKF [33]–[36]	VO	Shi Tomasi	Cross correlation	EKF	-	-	-	✓	-	✓	✓	-	Fair	Very Good	[37] ³
OKVIS [38], [39]	VO	BRISK	Descriptor	Local BA	-	-	-	-	-	✓	✓	✓	Good	Very Good	[40]
ROVIO [41], [42]	VO	Shi Tomasi	Direct	EKF	-	-	-	-	-	✓	✓	✓	Good	Very Good	[43]
ORBSLAM-VI [4]	SLAM	ORB	Descriptor	Local BA	DBoW2	DBoW2 PG+BA	-	✓	-	✓	-	-	Very Good	Very Good	-
VINS-Fusion [7], [44]	VO	Shi Tomasi	KLT	Local BA	DBoW2	DBoW2 PG	✓	-	✓	✓	✓	✓	Good	Exc.	[45]
VI-DSO [46]	VO	High grad.	Direct	Local BA	-	-	-	-	-	✓	-	-	Very Good	Exc.	-
BASALT [47]	VO	FAST	KLT (LSSD)	Local BA	-	ORB BA	-	-	-	-	✓	✓	Very Good	Exc.	[48]
Kimera [8]	VO	Shi Tomasi	KLT	Local BA	-	DBoW2 PG	-	-	-	-	✓	-	Good	Exc.	[49]
ORB-SLAM3 (ours)	SLAM	ORB	Descriptor	Local BA	DBoW2	DBoW2 PG+BA	✓	✓	✓	✓	✓	✓	Exc.	Exc.	[5]

ORB-SLAM 2 system overview

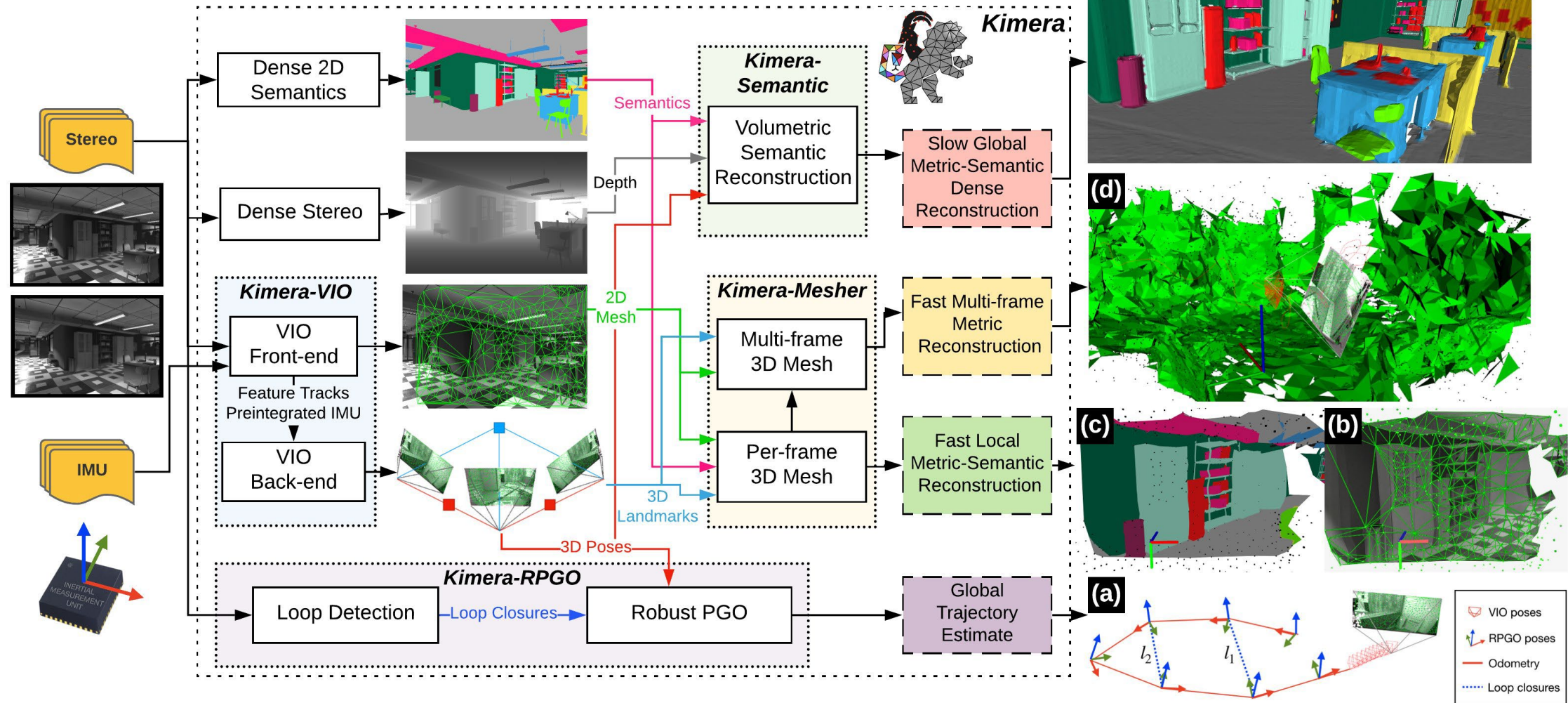


R. Mur-Artal and J. D. Tardos, "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras," IEEE Trans. Robot., pp. 1–8, 2017.

ORB-SLAM 3 system overview

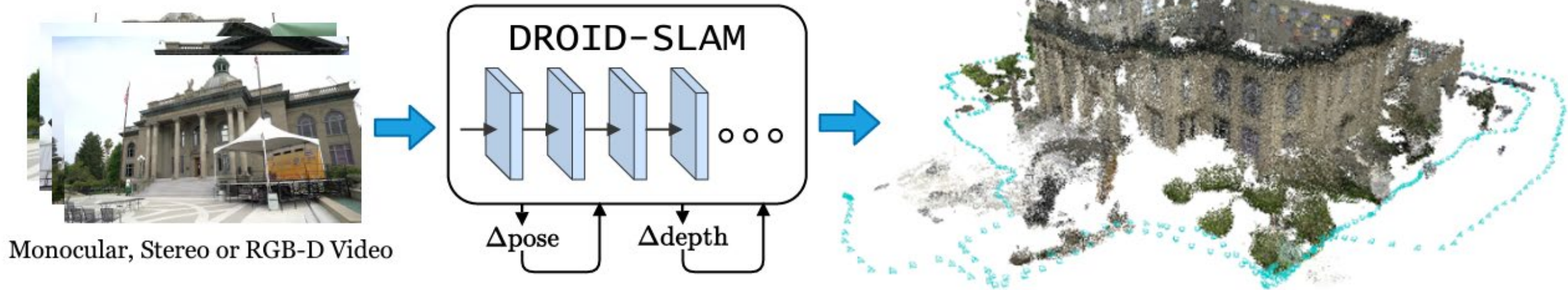


Kimera system overview



DROID-SLAM

Code: <https://github.com/princeton-vl/DROID-SLAM>

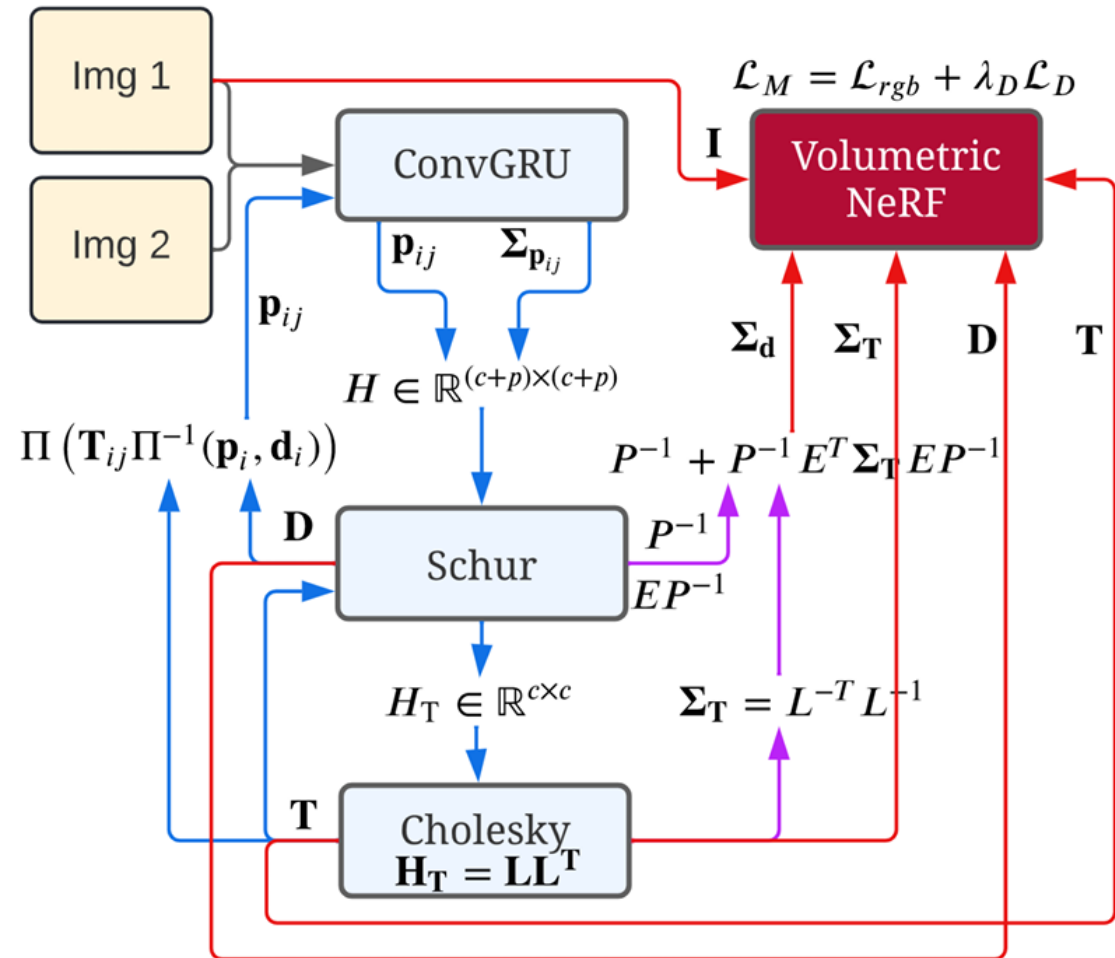


- DROID: Differentiable Recurrent Optimization-Inspired Design
- Indirect and dense!
- Learned update operator
- Local VO over fixed number of keyframes + global SLAM over all keyframes

Teed, Zachary, Deng, Jia. (2021). DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras
Arxiv. <https://arxiv.org/abs/2108.10869>

NeRF-SLAM

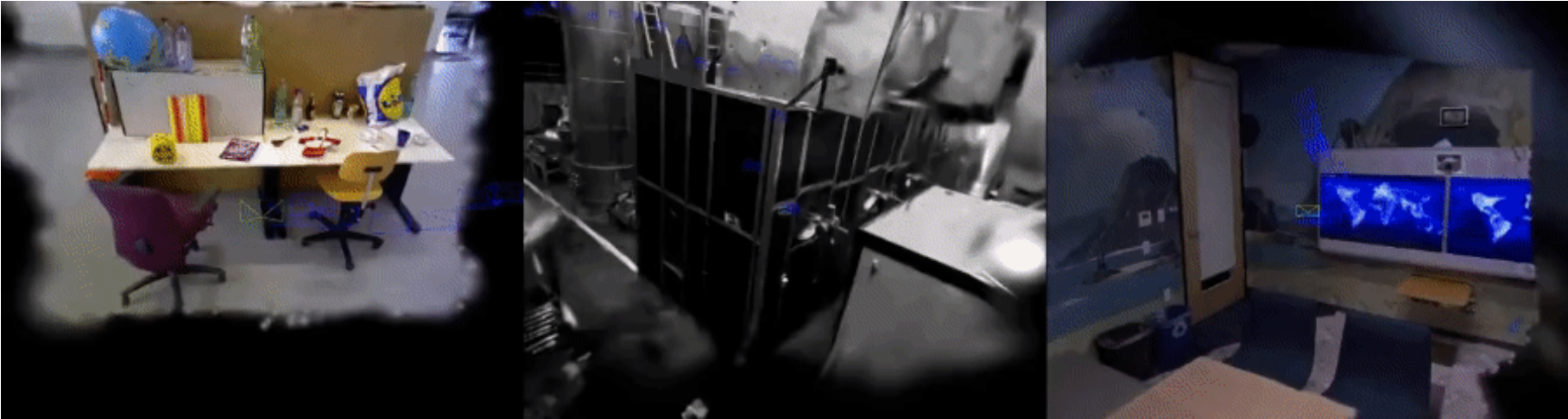
Code: <https://github.com/ToniRV/NeRF-SLAM>



Rosinol, Antoni, Leonard, John, Carlone, Luca (2022). NeRF-SLAM: Real-Time Dense Monocular SLAM with Neural Radiance Fields
Arxiv. <https://arxiv.org/abs/2210.13641>

Gaussian Splatting SLAM

Code: <https://github.com/muskie82/MonoGS>



H. Matsuki, R. Murai, P. H. J. Kelly, and A. J. Davison, 'Gaussian Splatting SLAM',
in 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA: IEEE, Jun. 2024, pp. 18039–18048. doi: 10.1109/CVPR52733.2024.01708.



OpenVINS

- <https://docs.openvins.com/>
- https://github.com/rpng/open_vins
- VIO based on a Multi-State Constraint Kalman Filter (MSCKF)
- SLAM backends
- Highly modular

