Machine Perception

Lecture 5: Localization and SLAM (Part I)

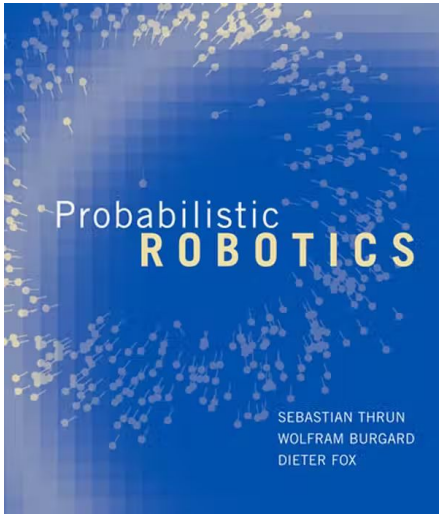
Piotr Skrzypczyński
Institute of Robotics and Machine Intelligence
Poznań University of Technology

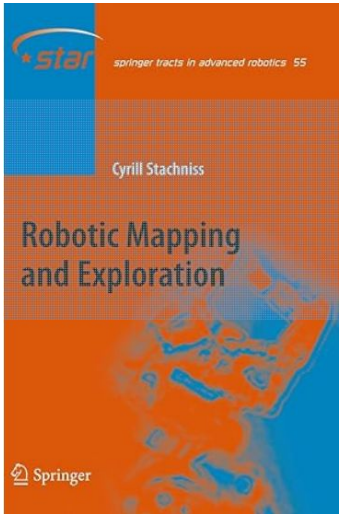
Lecture outline

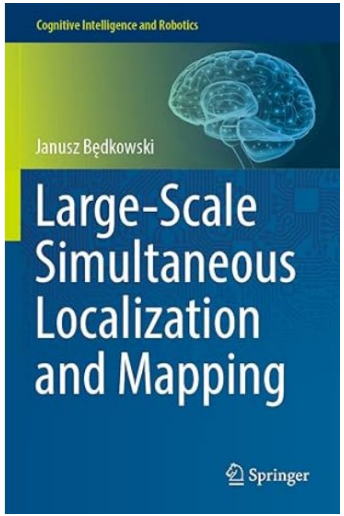
- Visual Odometry - definition, variants, role in SLAM.
- Main building blocks of Visual Odometry.
- Extending Visual Odometry to Visual SLAM.
- ORB-SLAM2 as an example of Visual SLAM.
- Applications of Visual SLAM.

Literature

1. S. Thrun, D. Fox, W. Burgard, Probabilistic Robotics, MIT Press, Cambridge, 2005.
2. C. Stachniss, Robotic Mapping and Exploration, Springer, 2009.
3. J. Będkowski, Large-Scale Simultaneous Localization and Mapping. Springer, 2022.

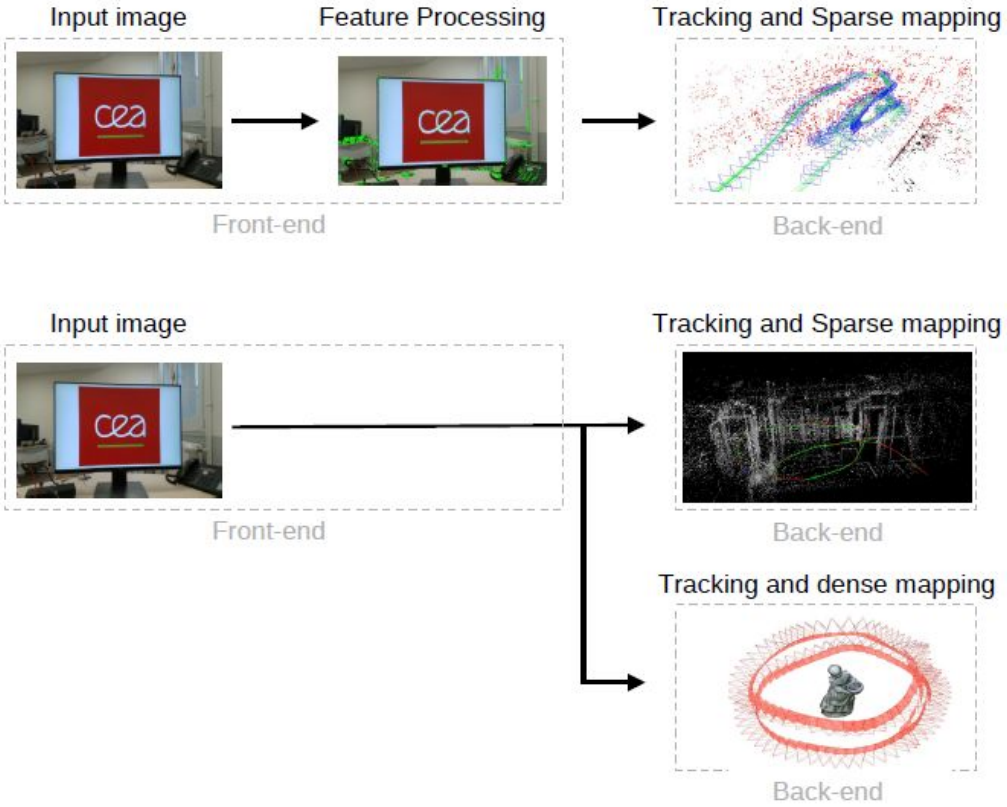






Introduction

- Simultaneous localization and mapping (SLAM) techniques are widely researched, since they allow the simultaneous creation of a map and the sensors' pose estimation in an unknown environment.
- Visual-based SLAM techniques play a significant role in this field, as they are based on a low-cost and small sensor system, which guarantees those advantages compared to other sensor based SLAM techniques.

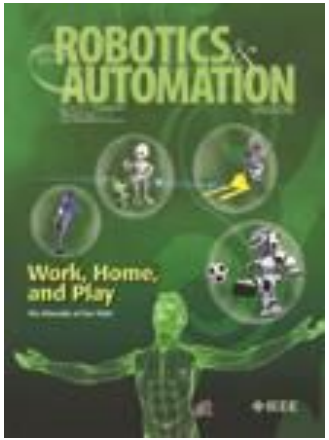


Visual Odometry

Visual Odometry (VO) is the process of incrementally estimating the pose of the vehicle by examining the changes that motion induces on the images of its onboard cameras.

Assumptions:

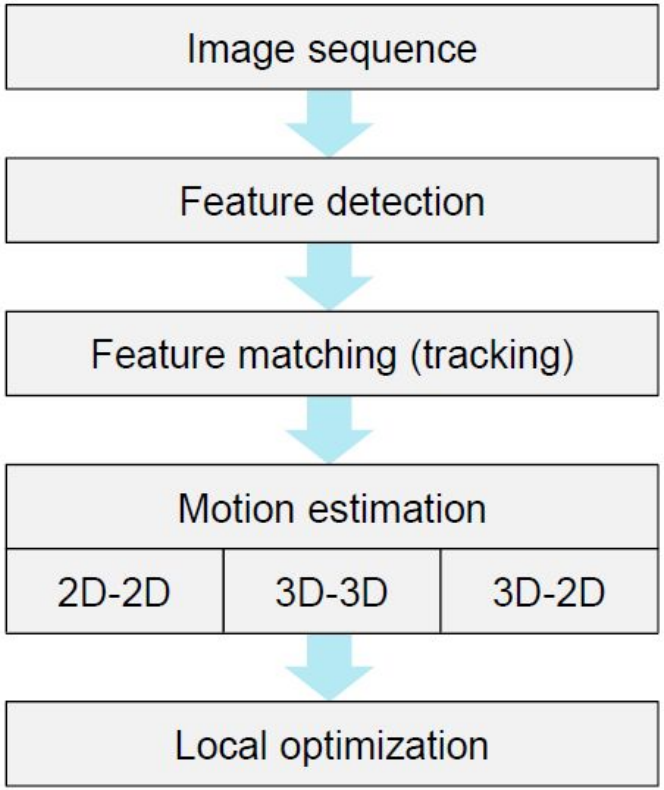
- Sufficient illumination in the environment
- Dominance of static scene over moving objects
- Enough texture to allow apparent motion to be extracted
- Sufficient scene overlap between consecutive frames



1. Scaramuzza, D., Fraundorfer, F., Visual Odometry: Part I - The First 30 Years and Fundamentals, IEEE Robotics and Automation Magazine, Volume 18, issue 4, 2011.
2. Fraundorfer, F., Scaramuzza, D., Visual Odometry: Part II - Matching, Robustness, and Applications, IEEE Robotics and Automation Magazine, Volume 19, issue 1, 2012.

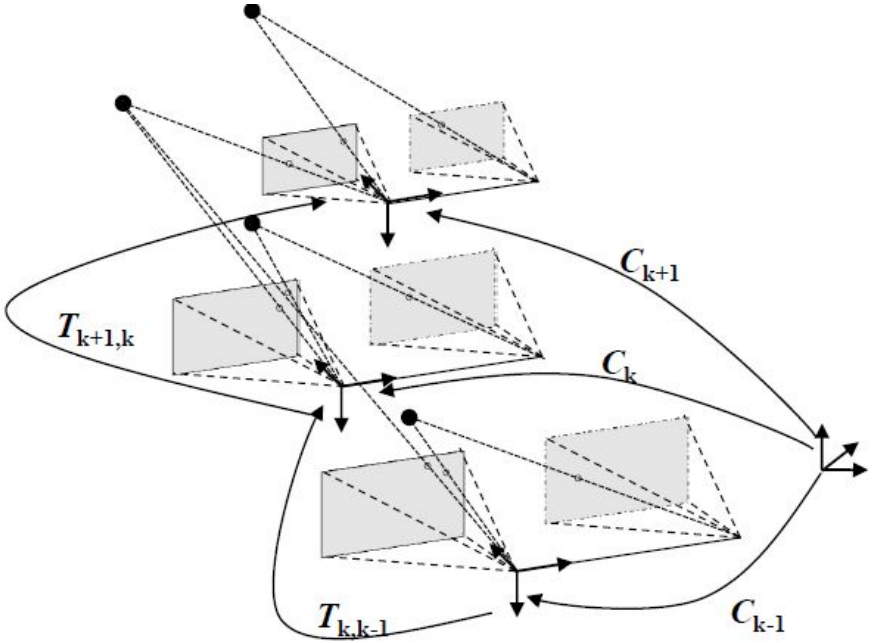
Visual Odometry

- Contrary to wheel odometry, VO is not affected by wheel slip in uneven terrain or other adverse conditions.
- More accurate trajectory estimates compared to wheel odometry.
- VO can be used as a complement to wheel odometry GPS inertial measurement units (IMUs).
- In GPS-denied environments, such as underwater and aerial, VO has utmost importance.

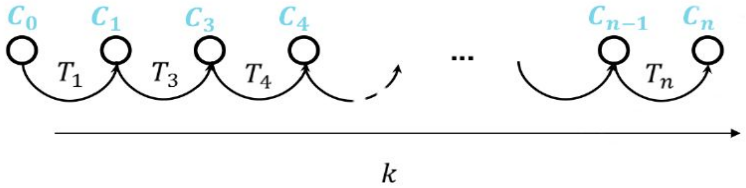


Visual Odometry

- Structure from Motion (SfM) is more general than VO and tackles the problem of 3D reconstruction of both the structure and camera. poses from **unordered** image sets
- In SfM the final structure and camera poses are typically refined with an offline optimization (i.e., bundle adjustment), whose computation time grows with the number of images.
- VO integrates frame-to-frame errors, thus generating a drift of the estimated trajectory.



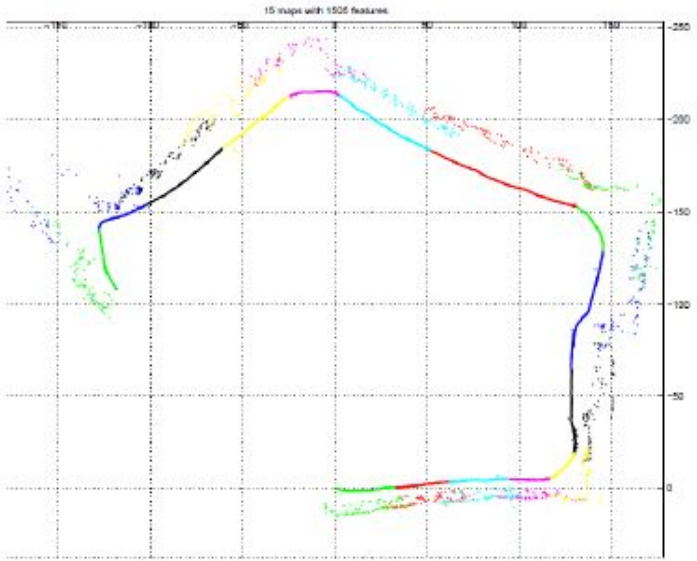
https://www.youtube.com/watch?v=kxtQqYLRaSQ



Visual Odometry and SLAM

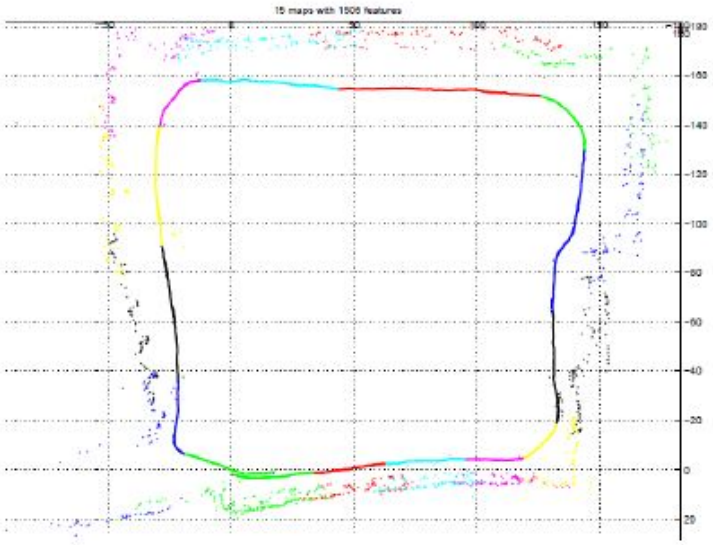
Simultaneous Localization and Mapping aims at estimating a global, consistent map and agent path. This is accomplished via identifying loop closures, and reducing the drift applying optimization (e.g. global bundle adjustment).

Visual Odometry aims at recovering the path incrementally, optimizing only over the last n camera poses (windowed bundle adjustment).



Visual Odometry and SLAM

- Simultaneous Localization and Mapping aims at estimating a global, consistent map and agent path. This is accomplished via identifying loop closures, and reducing the drift applying optimization (e.g. global bundle adjustment).
- Visual Odometry aims at recovering the path incrementally, optimizing only over the last n camera poses (windowed bundle adjustment).
- VO can be used as a building block of SLAM.

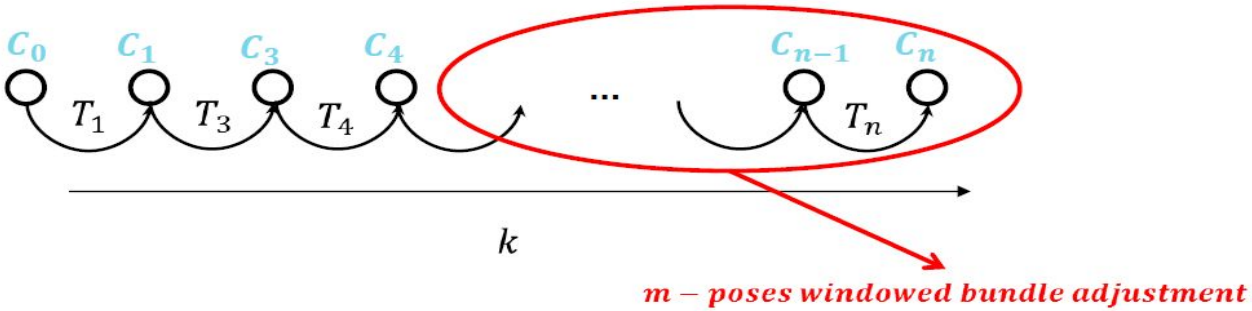


Visual Odometry

The main task in VO is to compute the relative transformations T_k from the images I_k and I_{k-1} and then to concatenate the transformations to recover the full trajectory $C_{0:n}$ of the camera.

This means that VO recovers the path incrementally, pose after pose.

An iterative refinement over last *m* poses can be performed after this step to obtain a more accurate estimate of the local trajectory.



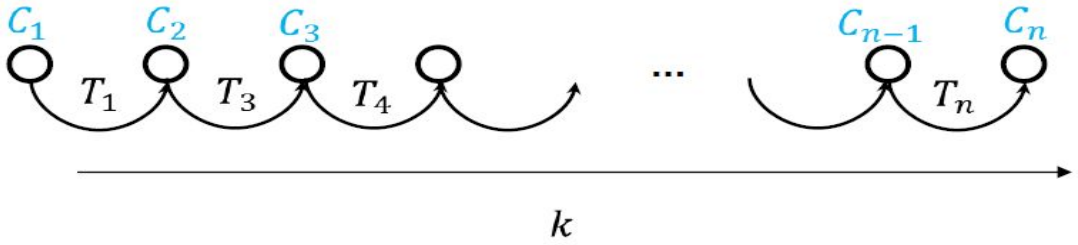
Visual Odometry

Motion estimation is the core computation step performed for every image in a VO system

It computes the camera motion T_k between the previous and the current image:

$$T_k = \begin{bmatrix} R_{k,k-1} & t_{k,k-1} \\ 0 & 1 \end{bmatrix}$$

By concatenation of all these single movements, the full trajectory of the camera can be recovered



Visual Odometry

Compute transformation $\mathbf{T}_k$ between two images I_{k-1} and I_k from two sets of corresponding features $\mathbf{f}_{k-1}$, $\mathbf{f}_k$.
Different algorithms depending on available sensor data:

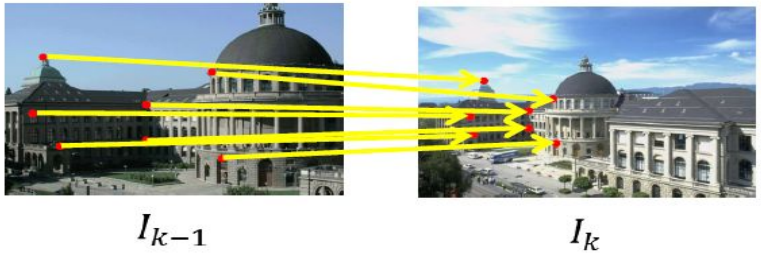
- 2-D to 2-D: works on $\mathbf{f}_{k-1}$, $\mathbf{f}_k$ specified in 2-D image coords
- 3-D to 3-D: works on $\mathbf{X}_{k-1}$, $\mathbf{X}_k$, sets of 3D points corresponding to $\mathbf{f}_{k-1}$, $\mathbf{f}_k$
- 3-D to 2-D: works on $\mathbf{X}_{k-1}$, set of 3D points corresponding to $\mathbf{f}_{k-1}$, and on $\mathbf{f}_k$ their corresponding 2-D re-projections on the image I_k

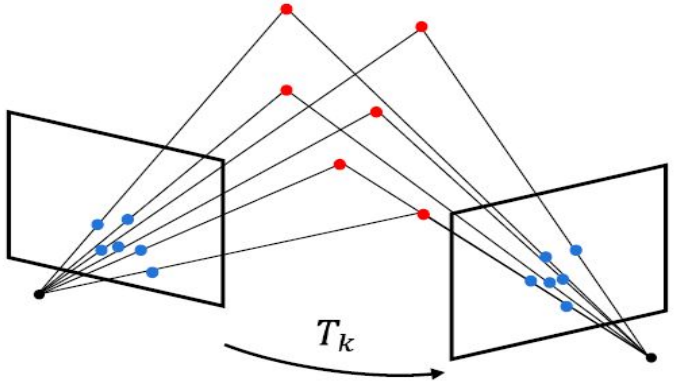
Type of correspondences	Monocular	Stereo
2D-2D	X	X
3D-3D		X
3D-2D	X	X

Visual Odometry (2D-to-2D)

- Both f_{k-1} and f_k are specified in 2D
- The minimal-case solution involves 5-point correspondences
- The solution is found by determining the transformation that minimizes the reprojection error of the triangulated points in each image

$$T_k = \begin{bmatrix} R_{k,k-1} & t_{k,k-1} \\ 0 & 1 \end{bmatrix} = \arg\min_{X^i, C_k} \sum_{i,k} \| p_k^i - g(X^i, C_k) \|^2$$





Visual Odometry (2D-to-2D)

$$p_1 = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \quad p_2 = \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix}$$ ***Image coordinates on the Unit sphere***

$$p_2^T \, E \, p_1 = 0$$ ***Epipolar constraint***

$$\mathrm{E} = [t]_\times R$$ ***Essential matrix***

$$[\boldsymbol{t}]_\times = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}$$

The Essential Matrix can be computed directly from the image coordinates (using SVD).

At least 5 points needed! [Kruppa, 1913]. The more points, the better!

The Essential Matrix can be decomposed into $\boldsymbol{R}$ and $\boldsymbol{t}$ (again using SVD)

Visual Odometry (2D-to-2D)

The Essential matrix can be computed from 5 point correspondences using [Nister'2003] algorithm (5-*point algorithm)*

The 5-p algorithm has become the standard for 2D-to-2D motion estimation, however, its implementation is not straightforward

A simple and straightforward solution for $n \geq 8$ noncoplanar points is the Longuet-Higgins' *8-p algorithm*, which is summarized here:

Let $p_1 = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}$, $p_2 = \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix}$ be the coordinates one feature correspondence

$$E = \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix} = \begin{bmatrix} e_{11} \\ \vdots \\ e_{33} \end{bmatrix}$$

$$p_2{}^{\mathrm{T}} E p_1 = 0 \Rightarrow [\boldsymbol{x_1 x_2}\ \boldsymbol{y_1 x_2}\ \boldsymbol{z_1 x_2}\ \boldsymbol{x_1 y_2}\ \boldsymbol{y_1 y_2}\ \ \boldsymbol{z_1 y_2}\ \ \boldsymbol{x_1 z_2}\ \boldsymbol{y_1 z_2}\ \ \boldsymbol{z_1 z_2}]E = 0$$

which can be solved with SVD

Visual Odometry (2D-to-2D)

Algorithm 1: VO from 2D-to-2D correspondeces

1 Capture new frame I_k
2 Extract and match features between I_{k-1} and I_k,
3 Compute essential matrix for image pair I_{k-1}, I_k
4 Decompose essential matrix into R_k and t_k, and form T_k
5 Compute relative scale and rescale t_k accordingly
6 Concatenate transformation by computing $C_k = C_{k-1}T_k$
7 Repeat from 1

Visual Odometry (3D-to-3D)

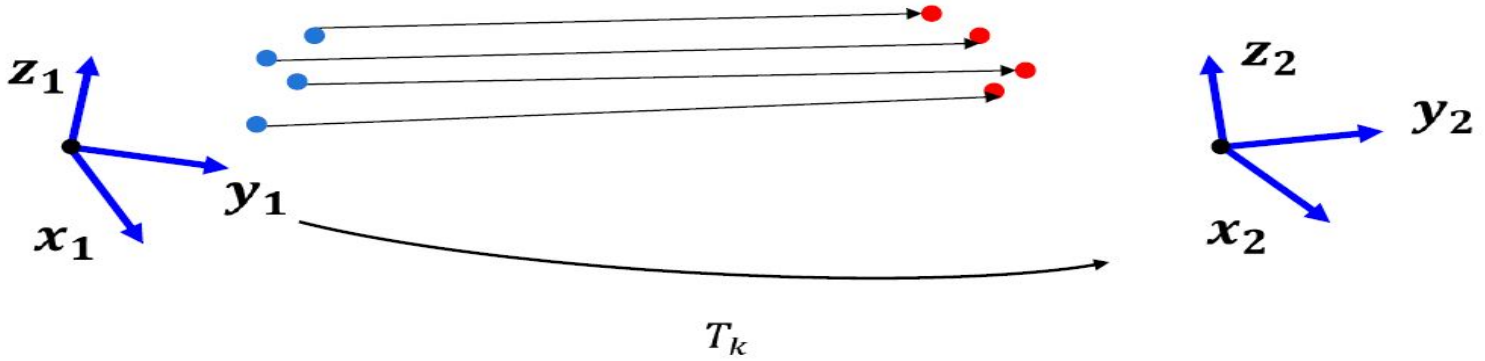
Both f_{k-1} and f_k are specified in 3D

To do this, it is necessary to triangulate 3D points (e.g. use a stereo camera)

The minimal-case solution involves 3 non-collinear correspondences

The solution is found by determining the aligning transformation that minimizes the 3D-3D distance

$$T_k = \begin{bmatrix} R_{k,k-1} & t_{k,k-1} \\ 0 & 1 \end{bmatrix} = \arg\min_{T_k} \sum_i ||\tilde{X}_k^i - T_k \tilde{X}_{k-1}^i||$$



Visual Odometry (3D-to-2D)

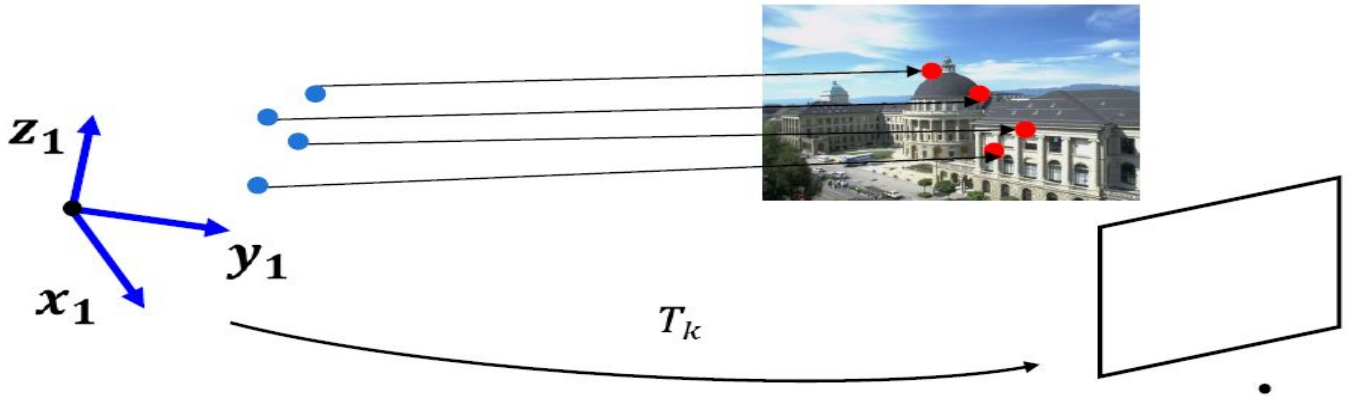
f_{k-1} is specified in 3D and f_k in 2D

This problem is known as *camera resection* or PnP (perspective from *n* points)

The minimal-case solution involves 3 correspondences

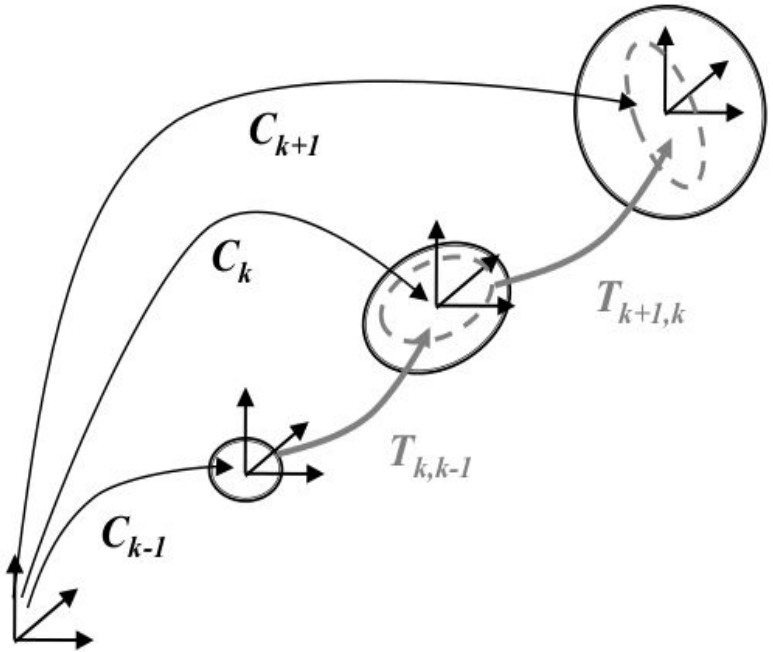
The solution is found by determining the transformation that minimizes the reprojection error

$$T_k = \begin{bmatrix} R_{k,k-1} & t_{k,k-1} \\ 0 & 1 \end{bmatrix} = \arg\min_{T_k} \sum_i \| p_k^i - \hat{p}_{k-1}^i \|^2$$



Visual Odometry (drift)

- The errors introduced by each new frame-to-frame motion accumulate over time.
- This generates a drift of the estimated trajectory from the real one.
- The uncertainty of the camera pose at $\mathbf{C}_k$ is a combination of the uncertainty at $\mathbf{C}_{k-1}$ (black solid ellipse) and the uncertainty of the transformation $\mathbf{T}_{k,k-1}$ (gray dashed ellipse)

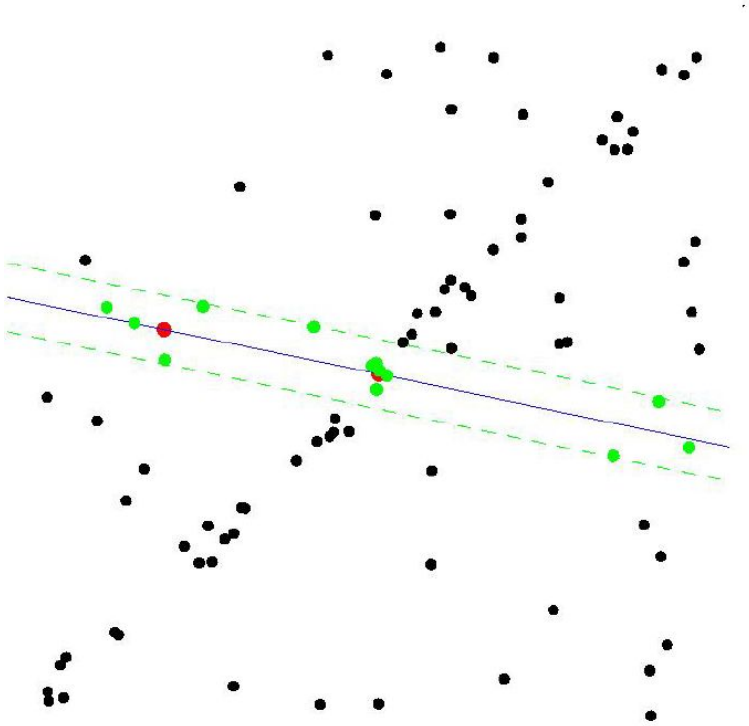


Visual Odometry (robust estimation)

- Matched points are usually contaminated by outliers, that is, wrong data associations.
- Possible causes of outliers are image noise, occlusions, blur, viewpoint changes, and illumination for which the feature detector or descriptor does not account for.
- For the camera motion to be estimated accurately, outliers must be removed
- Random Sampling Consensus (RANSAC).

Visual Odometry (robust estimation)

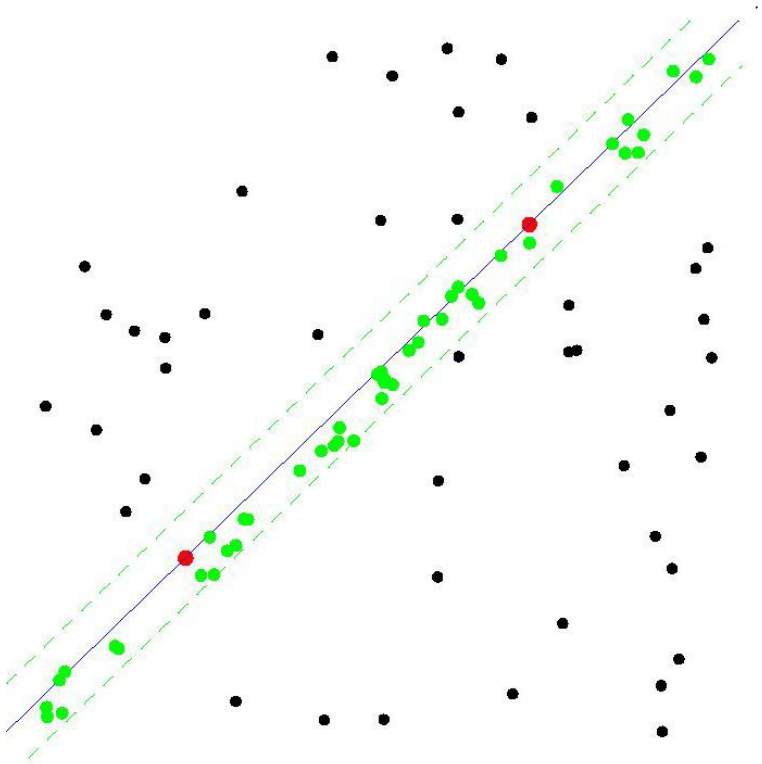
- Matched points are usually contaminated by outliers, that is, wrong data associations.
- Possible causes of outliers are image noise, occlusions, blur, viewpoint changes, and illumination for which the feature detector or descriptor does not account for.
- For the camera motion to be estimated accurately, outliers must be removed
- Random Sampling Consensus (RANSAC).



Visual Odometry (robust estimation)

Random Sampling Consensus (RANSAC)

1. Select sample of *n* points at random
2. Calculate model parameters that fit the data in the sample
3. Calculate error function for each data point
4. Select data that support current hypothesis
5. Repeat sampling



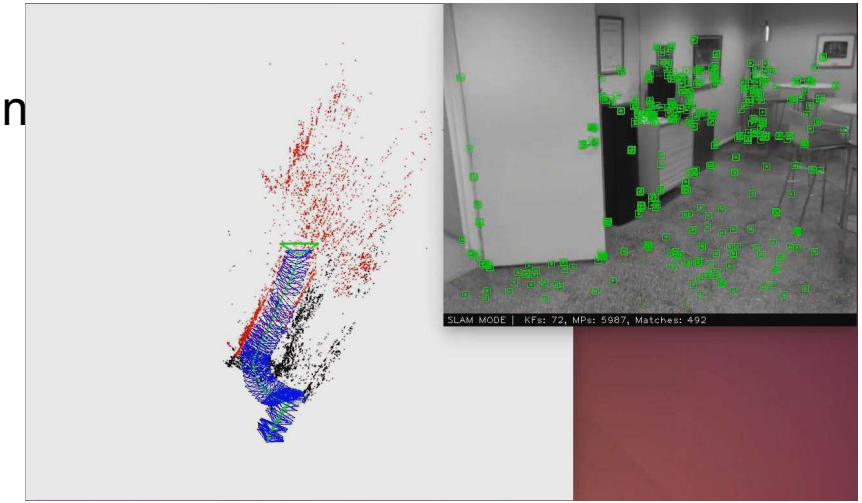
Visual Odometry

- In the stereo vision case, 3D-2D method exhibits less drift than 3D-3D method
- Stereo vision has the advantage over monocular vision that both motion and structure are computed in the absolute scale. It also exhibits less drift.
- When the distance to the scene is much larger than the stereo baseline, stereo VO degenerates into monocular VO
- Keyframes should be selected carefully to reduce drift
- Regardless of the chosen motion computation method, local bundle adjustment (over the last m frames) should be always performed to compute a more accurate estimate of the trajectory. After bundle adjustment, the effects of the motion estimation method are much more alleviated (as long as the initialization is close to the solution)

Visual SLAM

Simultaneous Localization and Mapping

- Mapping: continuously expanding an optimizing a consistent map while exploring the environment
- Localization (tracking): localization within the map(tracking the map in image frames)



Visual SLAM

Measurement model:

$$\mathbf{z}_i = h_i(X_i) + \eta, \qquad \eta \sim N(\mathbf{0}, \boldsymbol{\Sigma}_i)$$

Measurement prediction function:

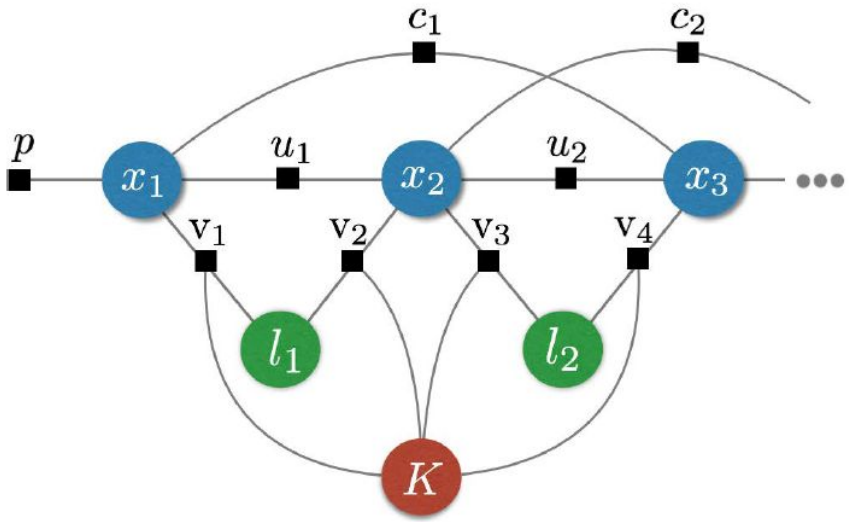
$$\hat{\mathbf{z}}_i = h_i(X_i)$$

Measurement likelihood:

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

MAP estimate:

$$X^{\mathrm{MAP}} = \underset{X}{\operatorname{argmin}} \sum_i \left\| h_i(X_i) - \mathbf{z}_i \right\|^2_{\boldsymbol{\Sigma}_i}$$

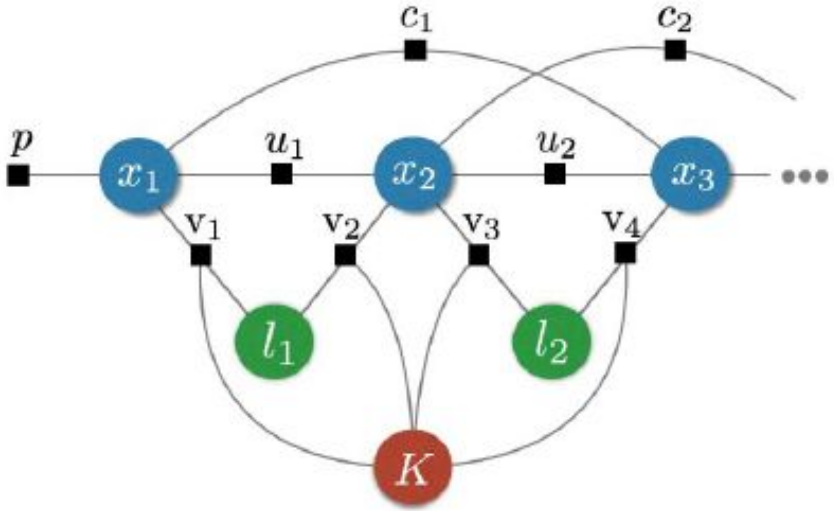


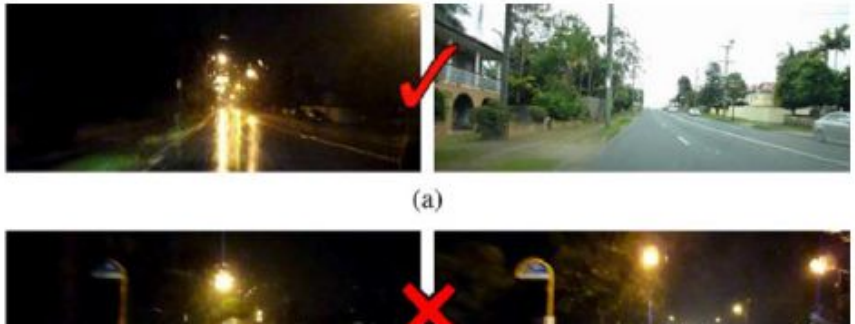
Cadena, C., et al. (2016). Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age. *IEEE Transactions on Robotics*, *32*(6), 1309–1332

Visual SLAM

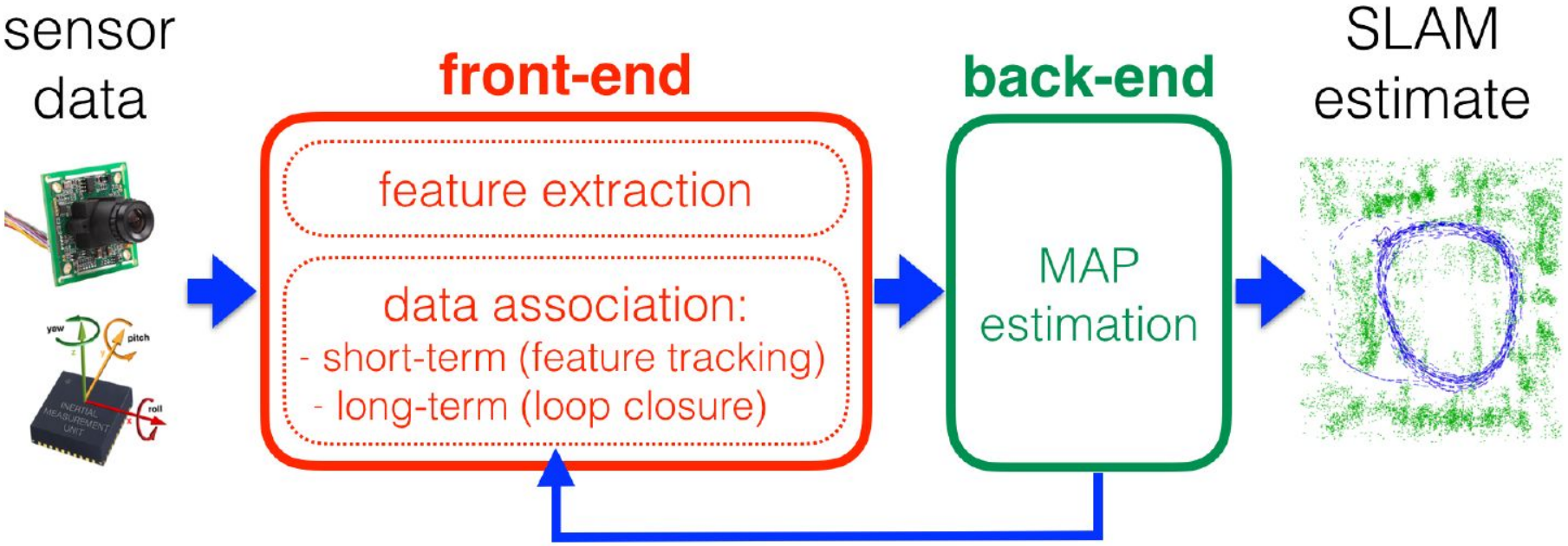
Components of Visual SLAM

- Short-term tracking
- Pose estimation given the map
- Keyframe proposals
- Long-term tracking
- Visual place recognition
- Loop closure detection over keyframes
- Mapping
- Building and optimizing the map over keyframes
- Data fusion





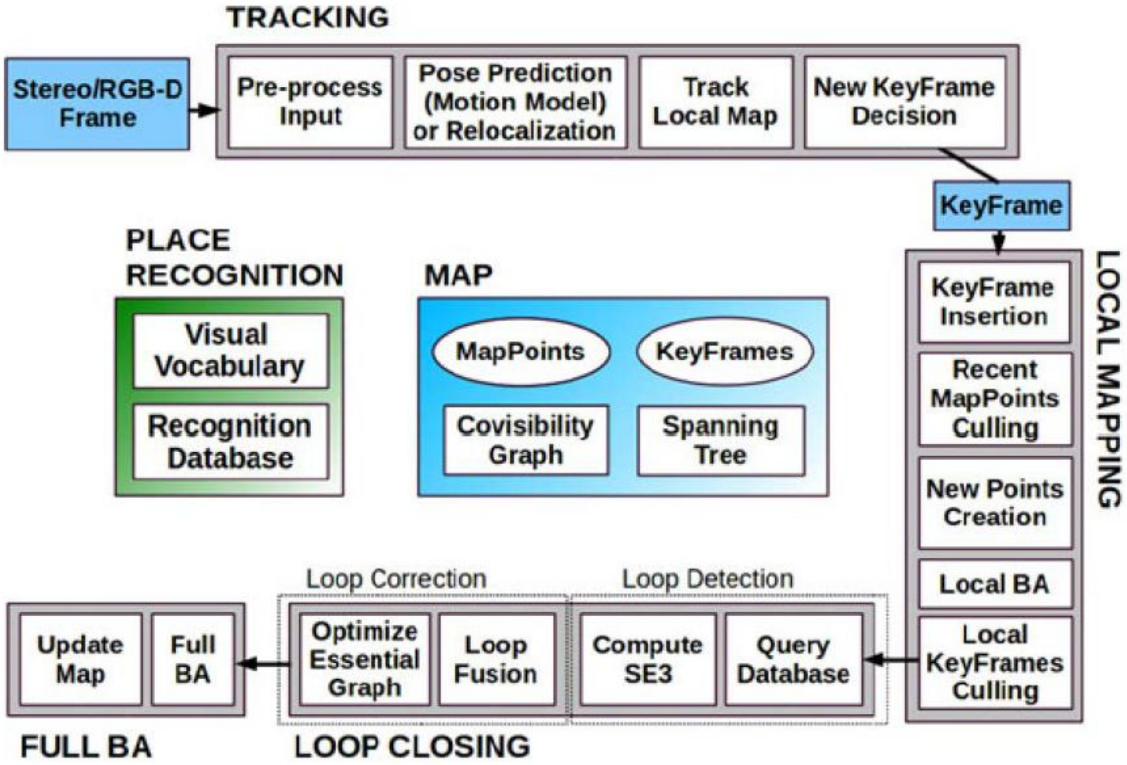
Visual SLAM



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

Visual SLAM (ORB-SLAM2)

Block diagram of ORB-SLAM2



- Use of the same features for all tasks. Real-time operation in large environments.
- Real-time loop closing based on the optimization of a pose graph.

Code: https://github.com/raulmur/ORB_SLAM2

Visual SLAM (ORB-SLAM2)

Short-term tracking

- FAST corners in grid cells at different scale levels with ORB descriptors

Initial pose estimation

- Tracking: guided search with constant velocity model
- Tracking lost: global relocalization

Track local map

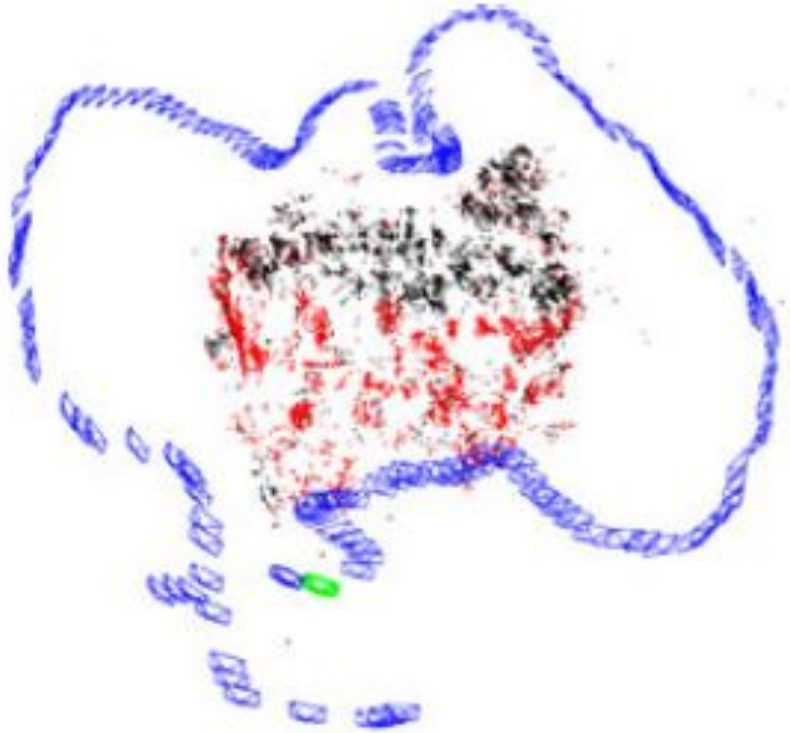
- Project local map and search for more correspondences
- Motion-only bundle adjustment

New keyframe decision

- Insert keyframes often to make tracking more robust to rotations

Map

- Keyframes (blue)
- Current frame (green)
- Map points (black)
- Active map points (red)



Visual SLAM (ORB-SLAM2)

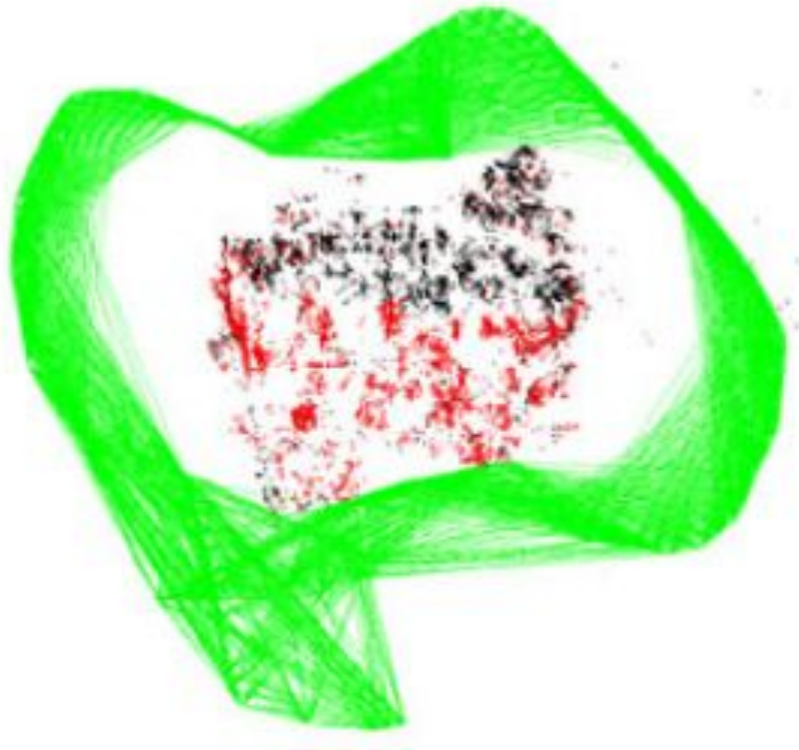
Short-term tracking
- FAST corners in grid cells at different scale levels with ORB descriptors

Initial pose estimation
- Tracking: guided search with constant velocity model
- Tracking lost: global relocalization

Track local map
- Project local map and search for more correspondences
- Motion-only bundle adjustment

New keyframe decision
- Insert keyframes often to make tracking more robust to rotations

Co-visibility graph
- Nodes: All keyframes
- Edges: Number of common map points (at least 15)



Visual SLAM (ORB-SLAM2)

Short-term tracking
- FAST corners in grid cells at different scale levels with ORB descriptors

Initial pose estimation
- Tracking: guided search with constant velocity model
- Tracking lost: global relocalization

Track local map
- Project local map and search for more correspondences
- Motion-only bundle adjustment

New keyframe decision
- Insert keyframes often to make tracking more robust to rotations

Spanning tree
- Connected subgraph of the co-visibility graph with minimal number of strong edges

Visual SLAM (ORB-SLAM2)

Short-term tracking

- FAST corners in grid cells at different scale levels with ORB descriptors

Initial pose estimation

- Tracking: guided search with constant velocity model
- Tracking lost: global relocalization

Track local map

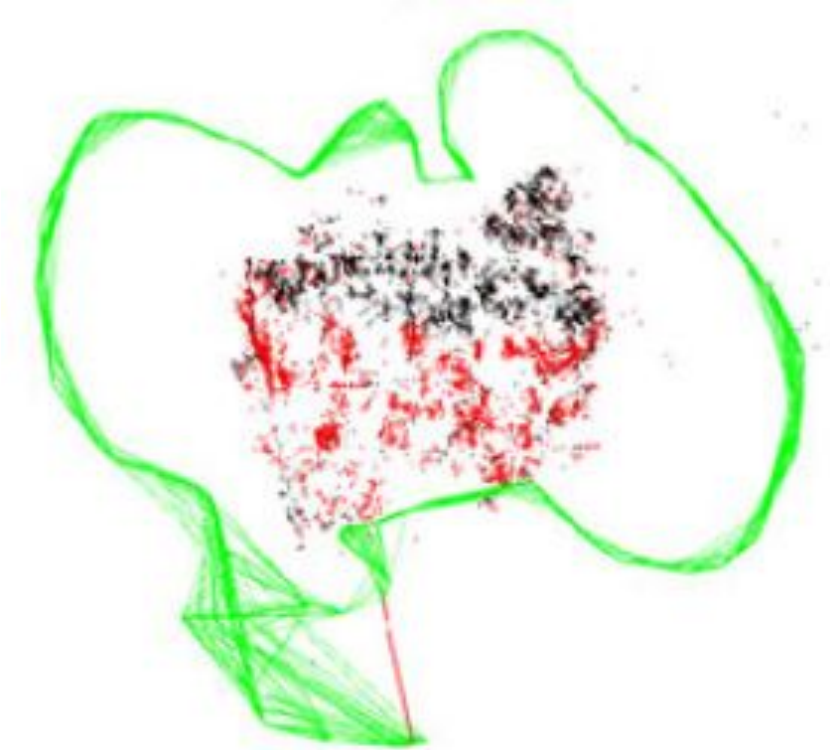
- Project local map and search for more correspondences
- Motion-only bundle adjustment

New keyframe decision

- Insert keyframes often to make tracking more robust to rotations

Essential graph

- Spanning tree
- Subset of edges from the co-visibility graph with high co-visibility (at least 100)
- Loop closure edges

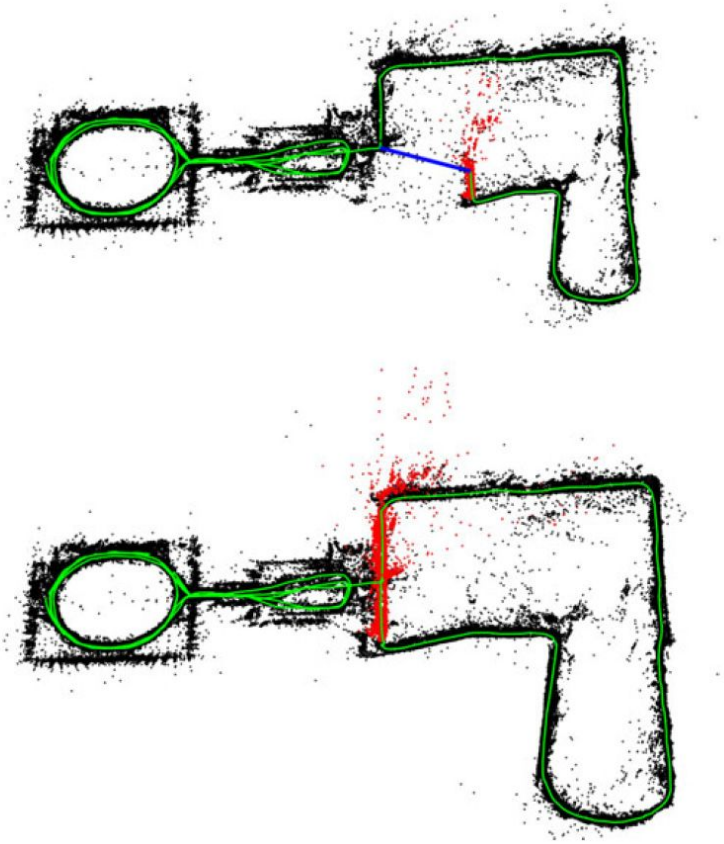


Visual SLAM (ORB-SLAM2)

Loop closing

1. Query recognition database (DBoW2) for keyframes with score higher than the threshold.
2. Loop fusion
   - Fuse map points
   - Insert new edges in the co-visibility graph

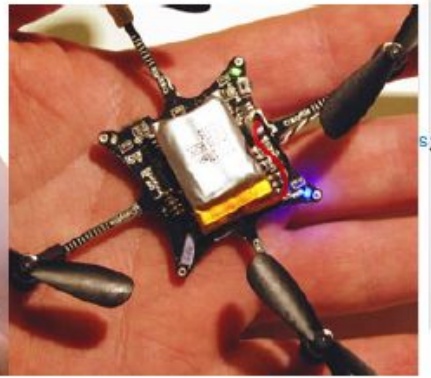
3. Essential graph optimization
   - Distribute the loop closing error along a pose graph over $\mathfrak{sim}(3)$
   - Transform each map point according to the correction of one of the keyframes that observes it



Visual SLAM - applications

Areas of Visual SLAM applications:

- Robotics (e.g. a mobile robot with a cheap camera)
- Agile robotics (e.g. drones)
- Autonomous driving
- Smartphones
- Wearables
- AR/VR: inside-out tracking, gaming

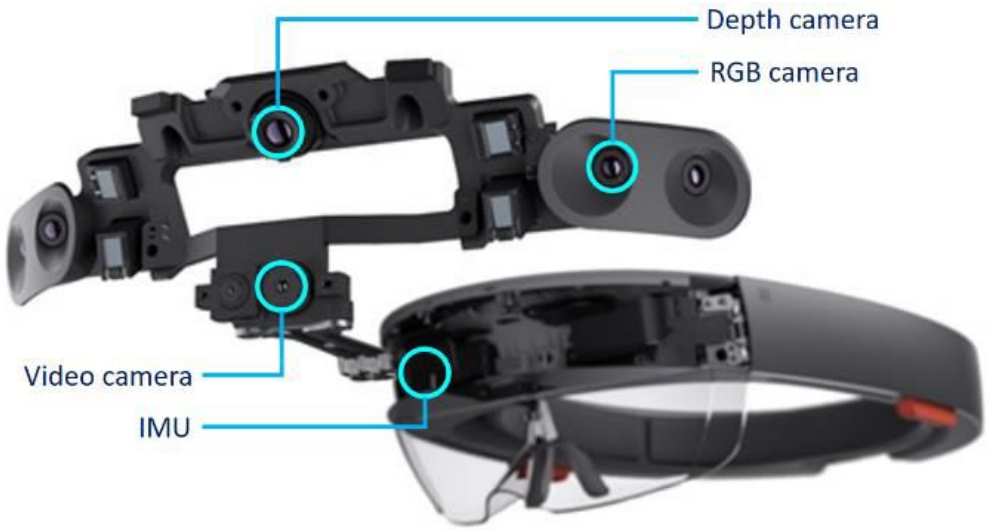




Visual SLAM - applications

- Robotics and Computer Vision market is exponentially growing.
- Robotic products, augmented reality and mixed reality apps/games, etc.
- Google (Project Tango, Google driverless car)
- Apple (acquisition of Metaio and Primesense)
- Dyson (funded Dyson Robotics Lab, Research lab at Imperial College in London)
- Microsoft (Hololens and its app marketplace)
- Magic Leap (funded by Google with $542M)
- Many apps related to machine learning and pattern recognition
- Quality open source systems: LSD-SLAM, ORB-SLAM, SVO, KinectFusion, ElasticFusion.
- Commercial products and prototypes: Google Tango, Hololens, Dyson 360 Eye, Roomba 980
- SLAM evolves into generic real-time 3D perception research

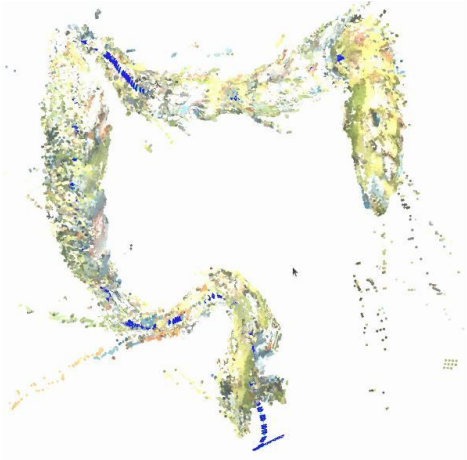
Visual SLAM - applications

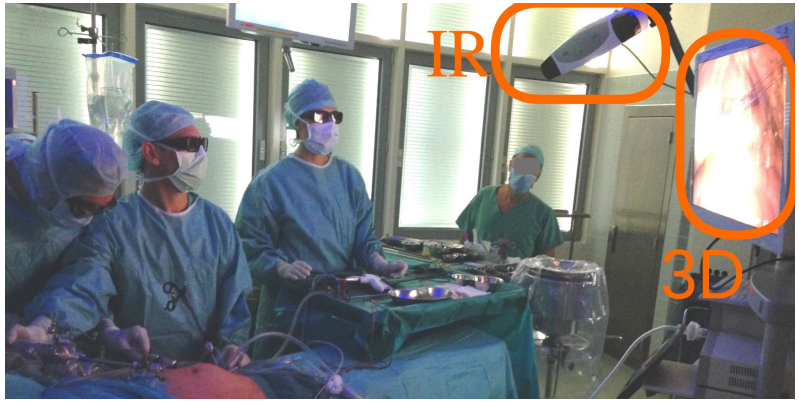
AR/VR headsets have built-in visual SLAM

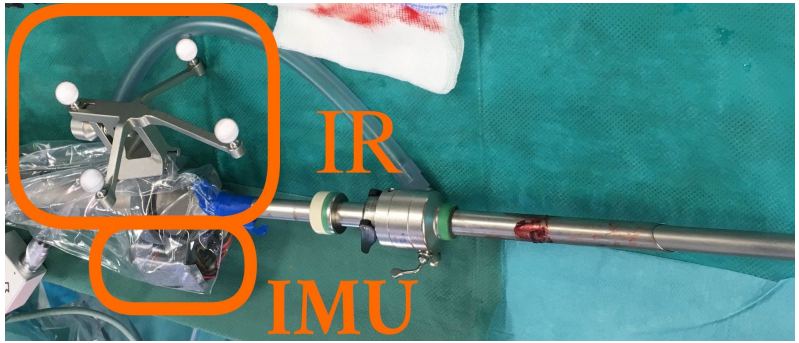


Visual SLAM - applications

Surgical navigation provides the necessary spatial information in computer-aided-surgery. Vision-based sensing has been proposed as a promising candidate for tracking and localisation application largely due to its ability to provide timely intra-operative feedback and contactless sensing.







Outcome of the lecture

- A brief review of the Visual localization and SLAM approaches - from a modern perspective (EKF SLAM is still OK, but not state of the art).
- More detailed presentation of the Visual Odometry principles: different perception - different solutions, robustness, optimization. Physical principles
- ORB-SLAM2 described in more detail as a good example of state of the art Visual SLAM system.
- Applications of modern VO/SLAM that go far beyond robotics.

IRIM

POZNAN UNIVERSITY OF TECHNOLOGY