Robot Vision

TTK4255

Lecture 12 – Localization and Mapping

Annette Stahl

(Annette.Stahl@ntnu.no)

Department of Engineering Cybernetics – ITK

NTNU, Trondheim

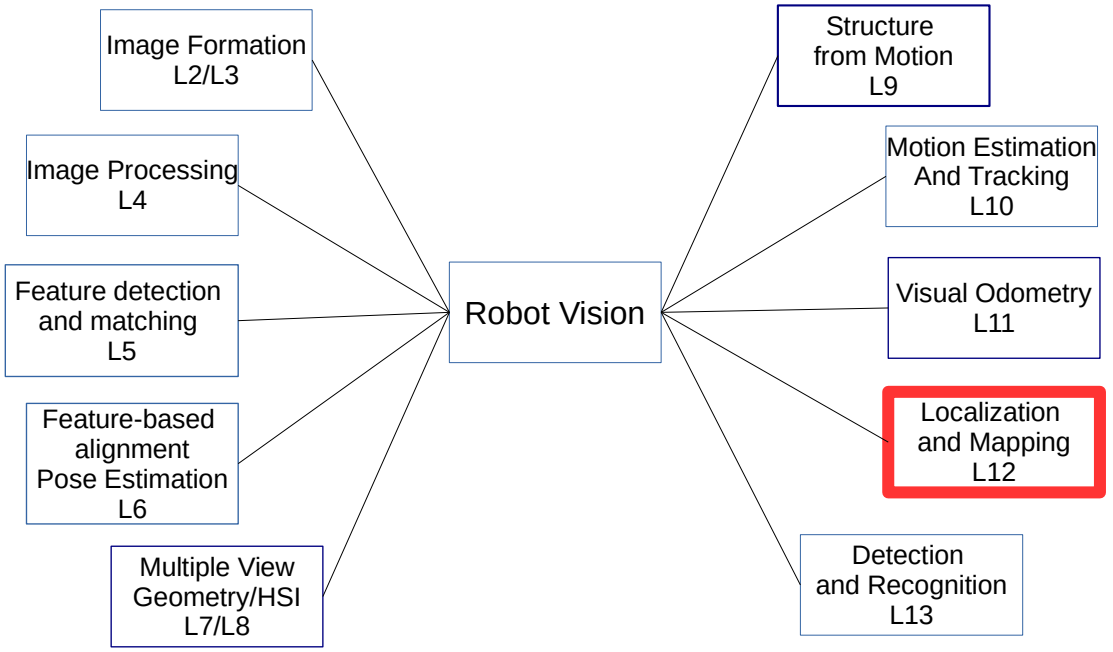
Spring Semester

23. March 2020

Lecture 12 – Localization and Mapping

Annette Stahl (Annette.Stahl@ntnu.no)

Simen Haugo (Simen.Haugo@ntnu.no)



Outline of the twelve lecture:

- SLAM
- Workflow
- Ingredients
- LSD – SLAM
- ORB - SLAM

Difference Visual Odometry – Visual SLAM

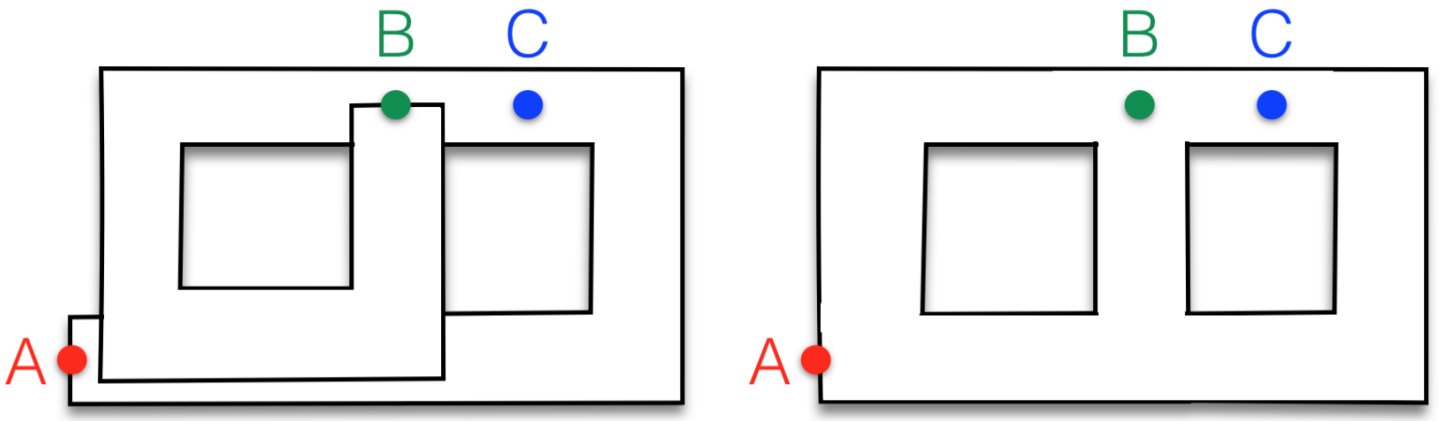
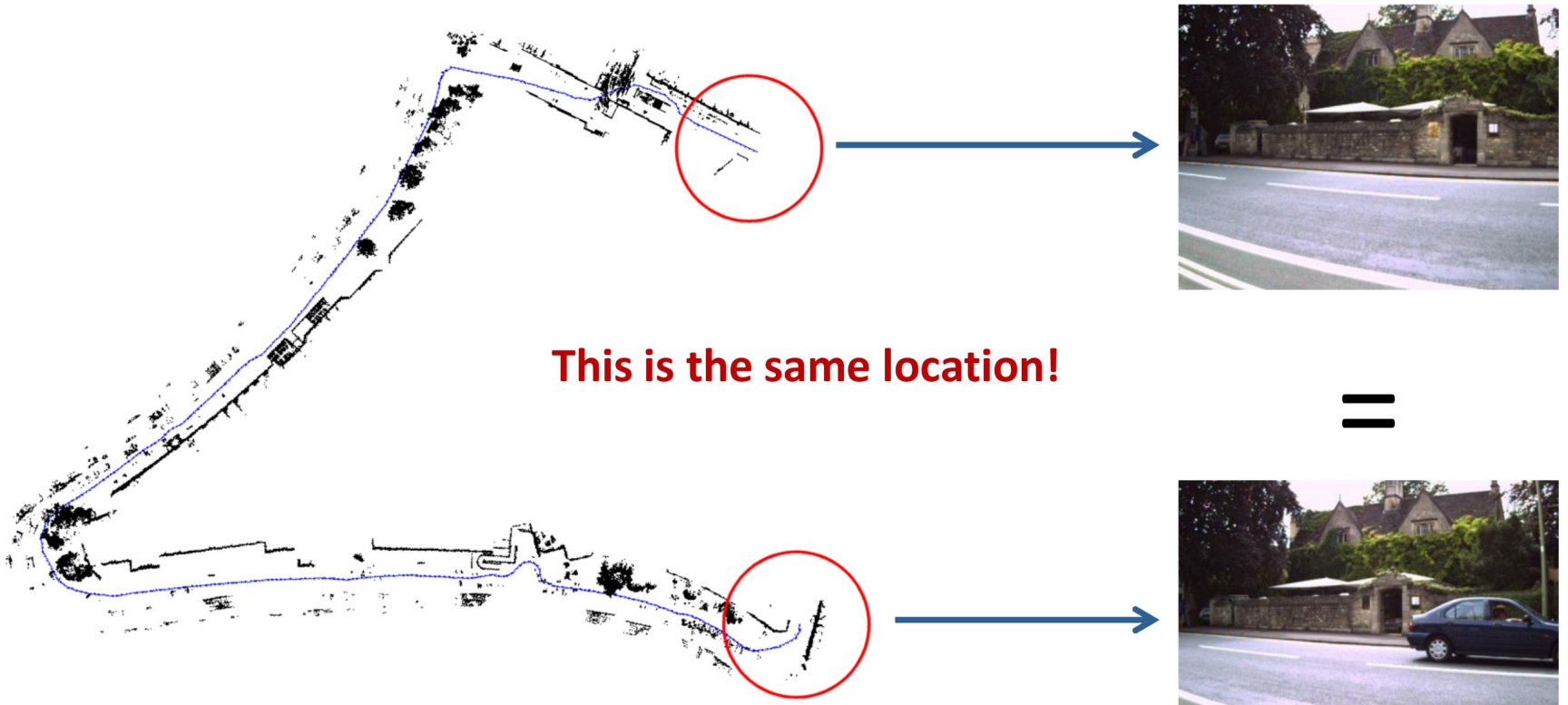


Fig. 1: Left: map built from odometry. The map is homotopic to a long corridor that goes from the starting position A to the final position B. Points that are close in reality (e.g., B and C) may be arbitrarily far in the odometric map. Right: map build from SLAM. By leveraging loop closures, SLAM estimates the actual topology of the environment, and "discovers" shortcuts in the map.

Motivation

Loop closing in Simultaneous Localization and Mapping



Sturm, J. (TUM)

Motivation

One of the most important tasks of **autonomous systems** of any kind is to **acquire knowledge about its environment and its location within it**. This is done by **taking measurements using various sensors** and then extracting meaningful information from those measurements.

SLAM

Simultaneous Localization and Mapping - SLAM

is the process by which a robot equipped with on-board sensors

builds a **model** (the **map) of the environment** that the senors are perceiving and,

at the same time,

uses the map to estimate its state (**its location)**:

- **Localization:** inferring location given a map
- **Mapping:** inferring a map given a location

Robot State: (camera) poses, robot velocity, sensor biases, camera intrinsics, 3D Geometry, 3D world point coordinates, etc.

Map is a representation of aspects of interest describing the environment (e.g. position of landmarks, obstacles, etc.)

SLAM Map

Reasons why to build a map:

- Map is required to support taks like, path planning or to provides an intuitive visualization for a human operator
- Map allows limiting the error committed in estimating the state of the robot.

→ **in absence of a map**, dead-reckoning

would quickly drift over time

increase in uncertainty

→ **having a map** the robot can reset

its localization error by revisiting

known areas (**loop closure**)

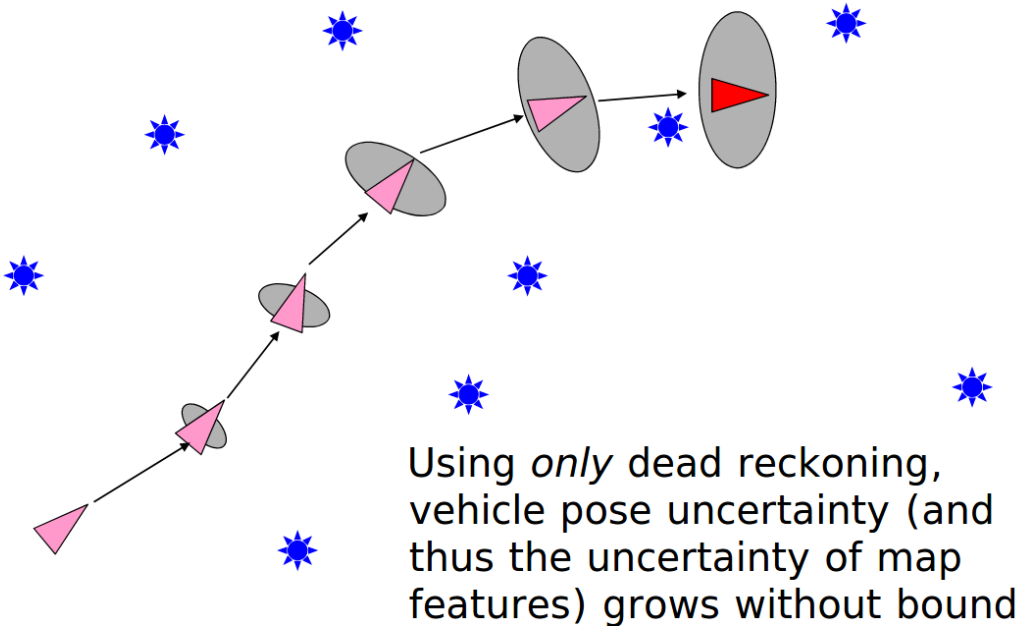


Image: Teller, Sigwart (MIT)

SLAM Principle Idea

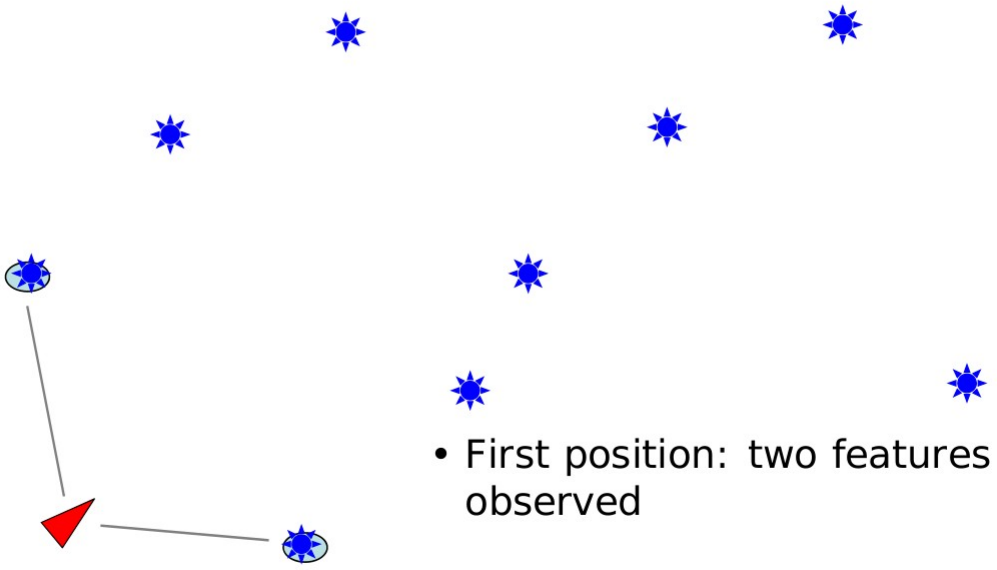


Image: Teller, Sigwart (MIT)

SLAM Principle Idea

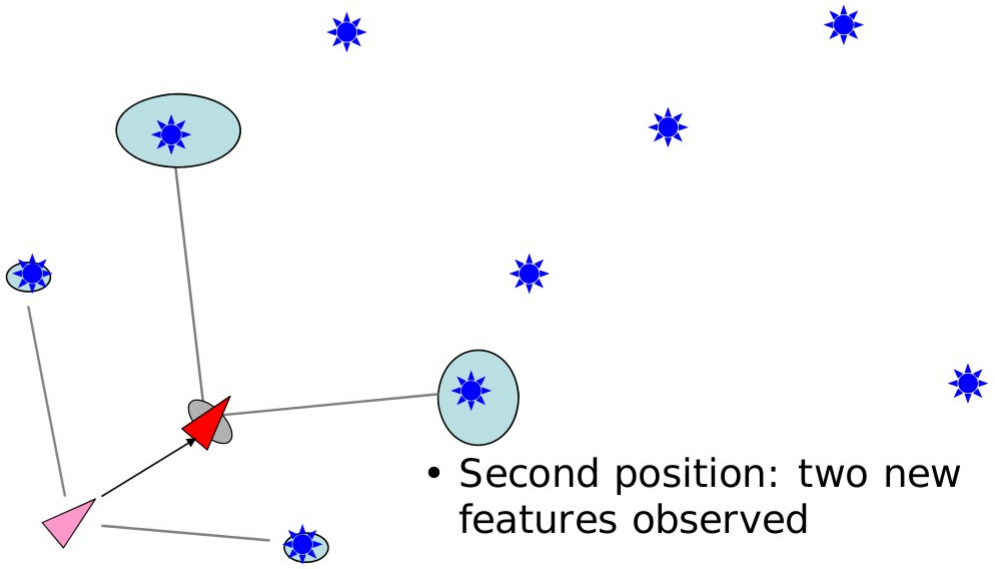
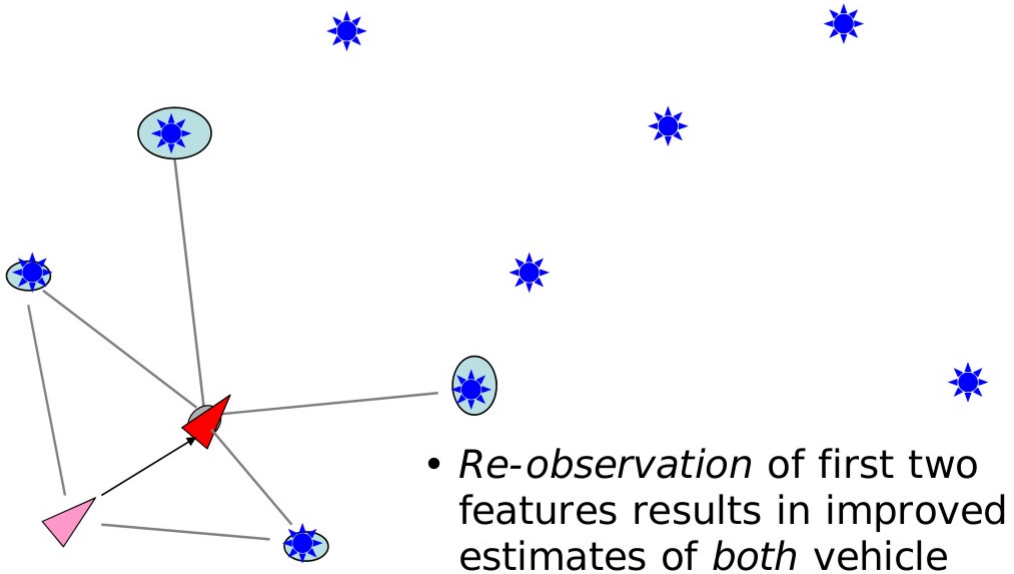


Image: Teller, Sigwart (MIT)

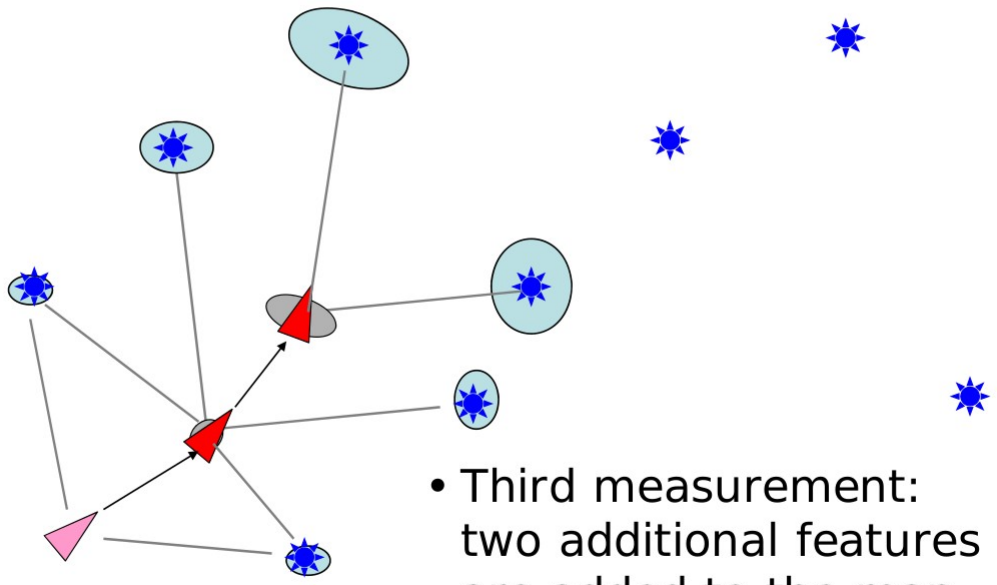
SLAM Principle Idea



- *Re-observation* of first two features results in improved estimates of *both* vehicle pose and features

Image: Teller, Sigwart (MIT)

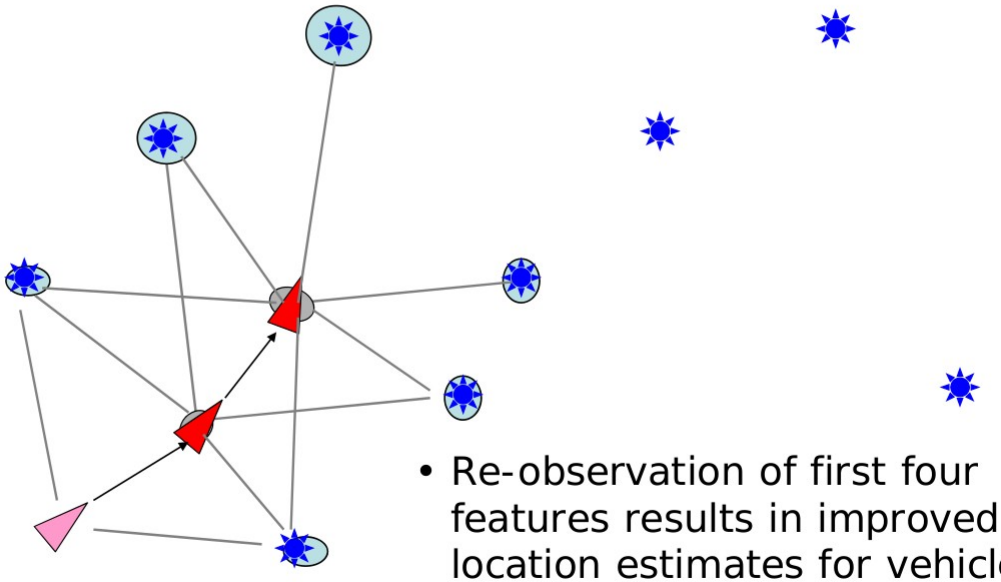
SLAM Principle Idea



- Third measurement: two additional features are added to the map

Image: Teller, Sigwart (MIT)

SLAM Principle Idea



- Re-observation of first four features results in improved location estimates for vehicle poses and all map features

Image: Teller, Sigwart (MIT)

SLAM Principle Idea

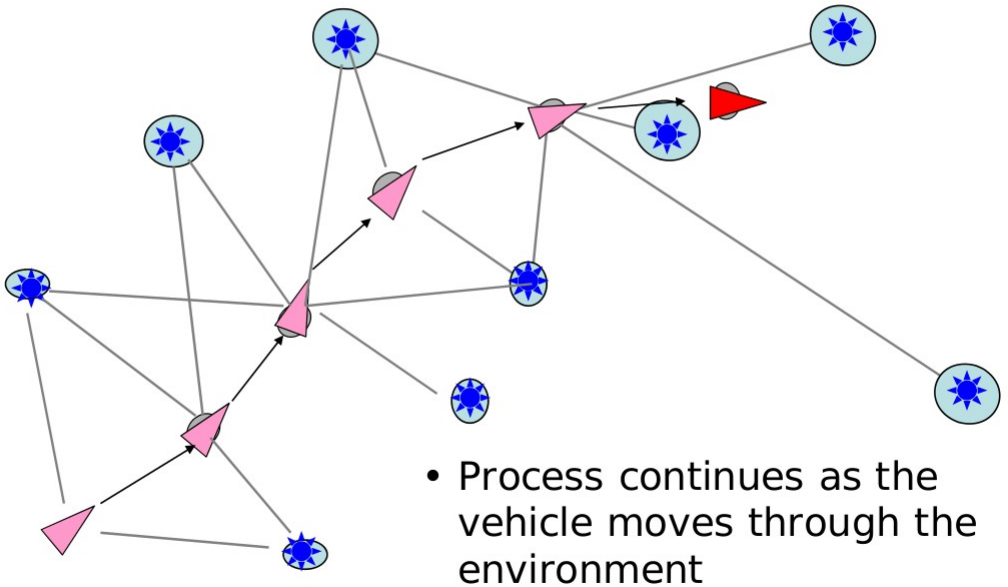


Image: Teller, Sigwart (MIT)

SLAM

Three required components for SLAM:

- Exteroceptive sensor to **measure external state**.
- Link between **measurement and internal state**.
- Link between **individual measurements, or data association**. Multiple measurements observing the same landmarks. There need to be a link between **internal state and external landmarks** through the measurement.

Given enough such links and measurements one can simultaneously solve for both localization and map representation. This can be done with different sensors. One of the most used is LiDAR, another popular sensor for SLAM is camera. Then it is often referred to as visual SLAM.

Exteroceptive sensors acquire information from the robot's environment; e.g. distance measurements, light intensity, sound amplitude. Hence exteroceptive sensor measurements are interpreted by the robot in order to extract meaningful environmental features.

SLAM

http://webdiis.unizar.es/~raulmur/orbslam/

VSLAM

Visual Simultaneous Localization and Mapping - SLAM

is the process by which a robot builds a **map of the environment** and,

at the same time,

uses the map to **compute its location**

using visual sensor input (image frames):

- **Localization:** inferring location given a map
- **Mapping:** inferring a map given a location

Simultaneous

- **mapping:** Continuously creating a model of the environment (map), (= expanding and optimizing a consistent map while exploring the environment)
- **localization:** Localization within the map (= estimating the state of the robot moving within the map) using visual sensor input (image frames)

VSLAM

Visual sensors:

Cameras have very fast and accurate place recognition capabilities:

- for finding **data association** between successive images used for **tracking**,
- for **long distance place recognition** used for **loop closures**.

→ Vision based SLAM methods are generally divided by two dimensions

- **Direct vs indirect** methods
- **Sparse vs dense** methods

Direct vs Indirect Methods

Discussion on how to perform **data association** (i.e. given an environment map and a set of sensor observations - associate observations with map elements)

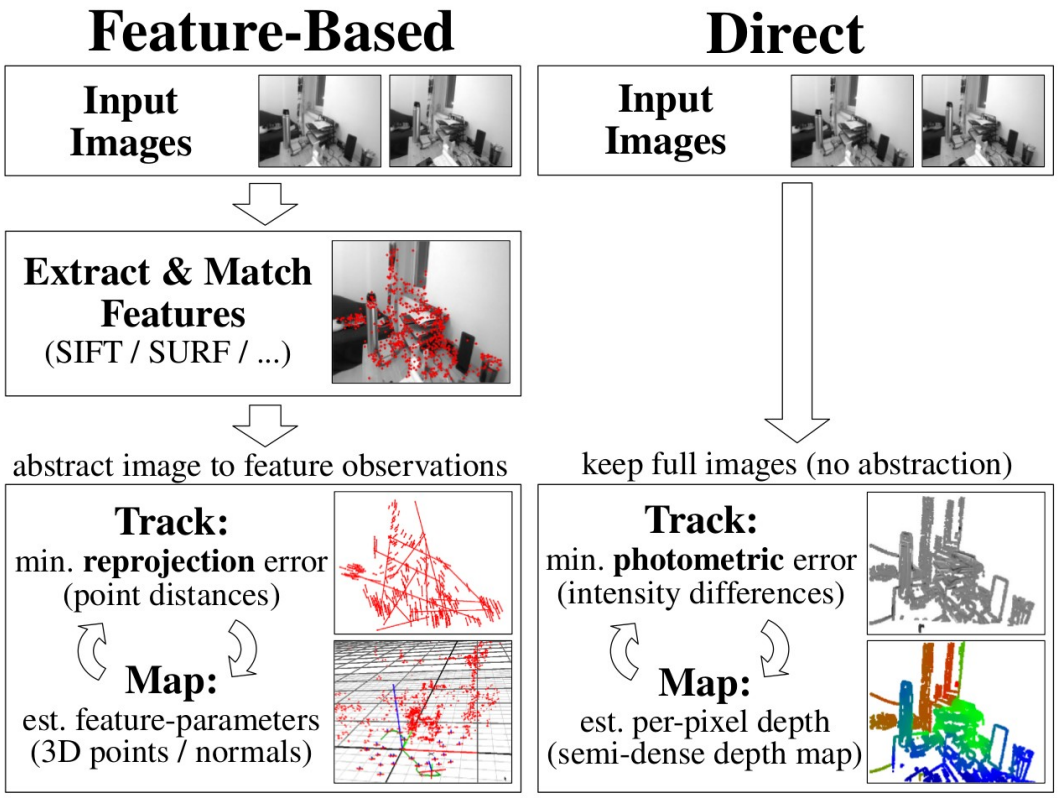


Figure 2: Feature based methods abstract images to feature observations and discard all other information. In contrast, the proposed direct approach maps and tracks directly on image intensities: this allows to (1) use all information, including e.g. edges and (2) directly obtain rich, semi-dense information about the geometry of the scene.

Direct: image intensities are compared directly; minimization of photometric error

Indirect (Feature-based): image intensities compared indirectly by first extracting image features; minimization of geometric error

Semi-Dense Visual Odometry for AR on a Smartphone (T. Schöps, J. Engel and D. Cremers), In International Symposium on Mixed and Augmented Reality, 2014.

https://vision.in.tum.de/research/vslam/lsdslam

Direct vs Indirect Methods

Indirect methods

first extract an **intermediate geometrical representation** of the scene, such as key-point correspondences or optical flow vectors, and use these to **optimise a geometric error**. Because these methods are based on some kind of feature extraction step, they are also often called **feature-based methods**. Indirect methods provide robustness to photometric and geometric distortions, but come at the price of higher computational cost and are dependent on the feature extraction step to work.

Direct methods

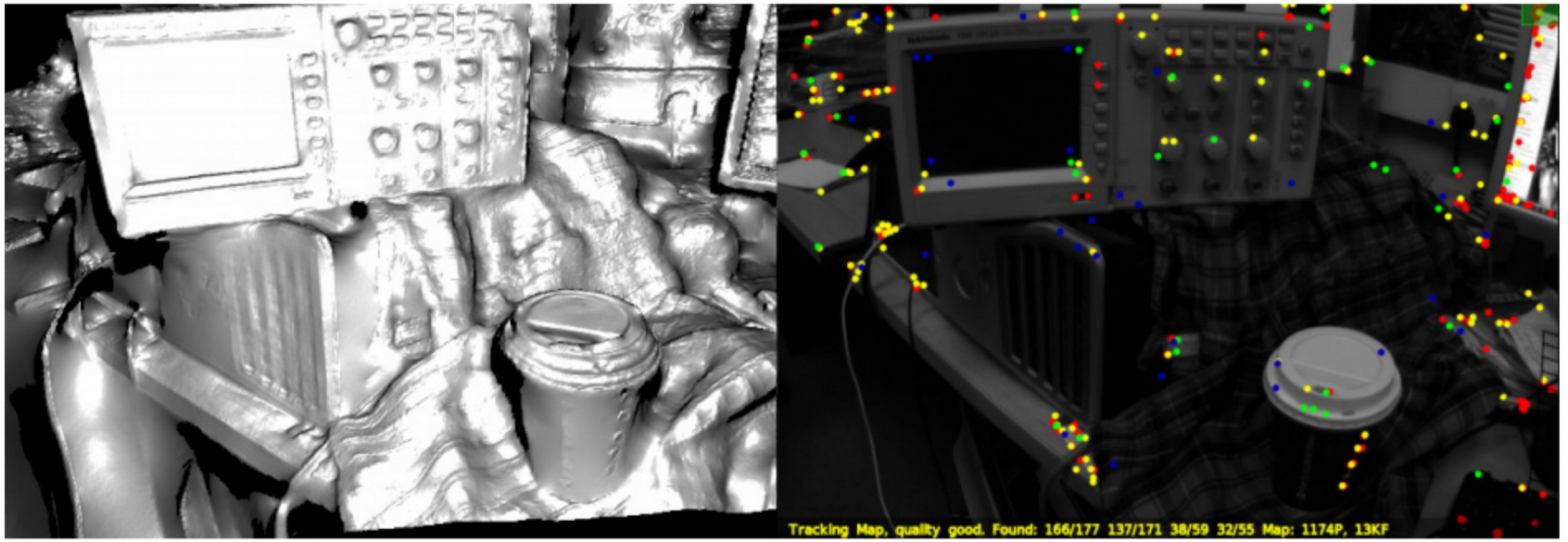
skip the feature extraction step, and use the **image intensity values directly** to **optimise a photometric error**. Direct methods do not require that geometrical primitives are recognisable by themselves, but can **sample across all parts of the image**. This makes direct methods more robust to effects such as motion blur and better suited in environments that are sparsely textured, but theyare vulnerable to photometric and geometric distortions

Dense vs Sparse Methods

Discussion on how dense the map should be.

Dense methods requires more computational power and needs to use assumptions such as smoothness and colour conservation to simplify the problem.

Sparse methods don’t need to make assumptions and can run in real time on a relative small CPU, but the resulting map is not directly usable for object avoidance or interactions.



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.

Map Representations

Map: A model of the environment that lets us

- limit the localization error by recognizing previously visited areas
- support other tasks, such as obstacle avoidance and path planning

Map Representations:

- Feature-based metric maps (ORB-SLAM)
- Dense metric maps (DTAM: Dense Tracking and Mapping in Real-
- Topological maps (FAB-MAP)
- Topological-metric maps (Visual Teach & Repeat)

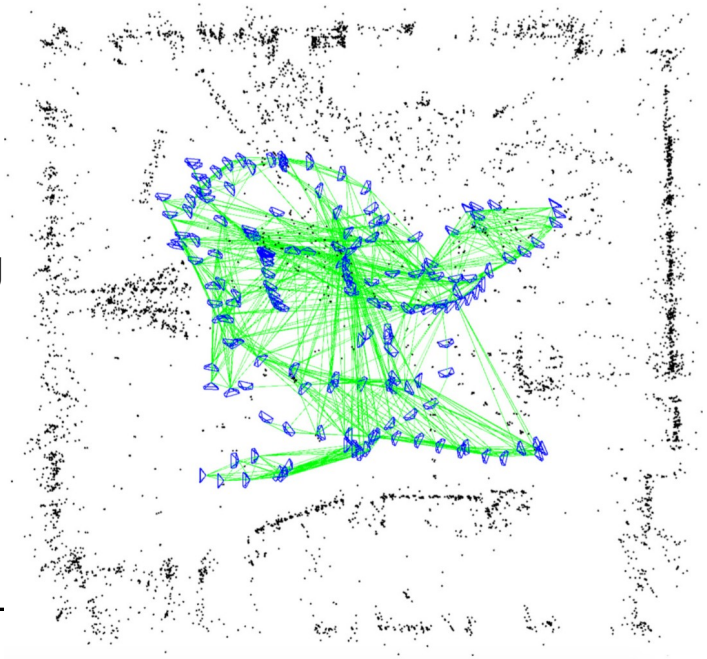


Image: Cadena, C., et al. (2016). Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age. *IEEE Transactions on Robotics*, *32*(6), 1309–1332

- 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.
- 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
- 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
- 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.

Components of a SLAM system

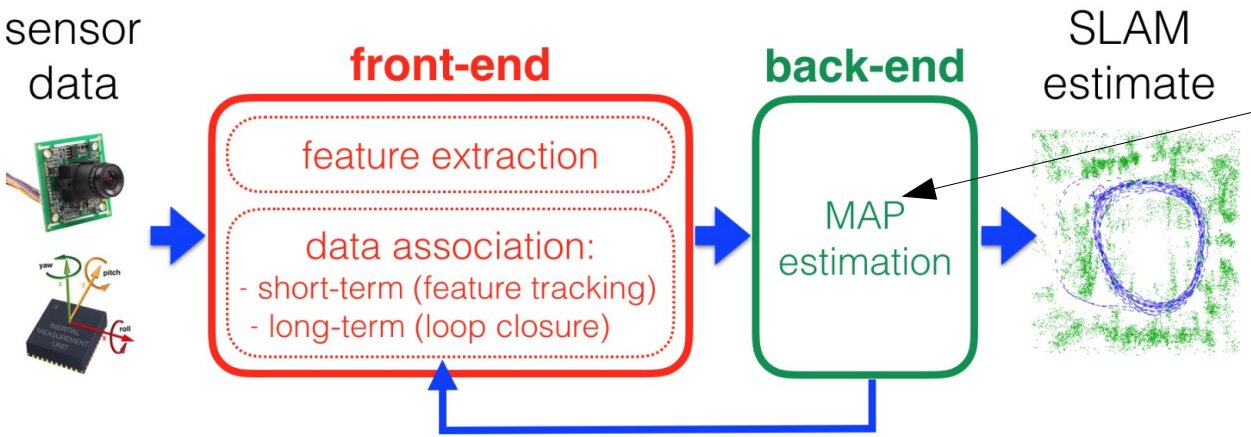


Fig. 2: Front-end and back-end in a typical SLAM system. The back-end can provide feedback to the front-end for loop closure detection and verification.

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

- Short-term tracking
  - Pose estimation given the map
  - Keyframe proposals

- Long-term tracking
  - Visual place recognition
  - Loop closure detection over keyframes

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

- Mapping
  - Building and optimizing the map over keyframes
  - Data fusion

Recall: Structure from Motion/Visual Odometry

Methods which tackle the problem of structure and motion estimation (also VO) in several steps:

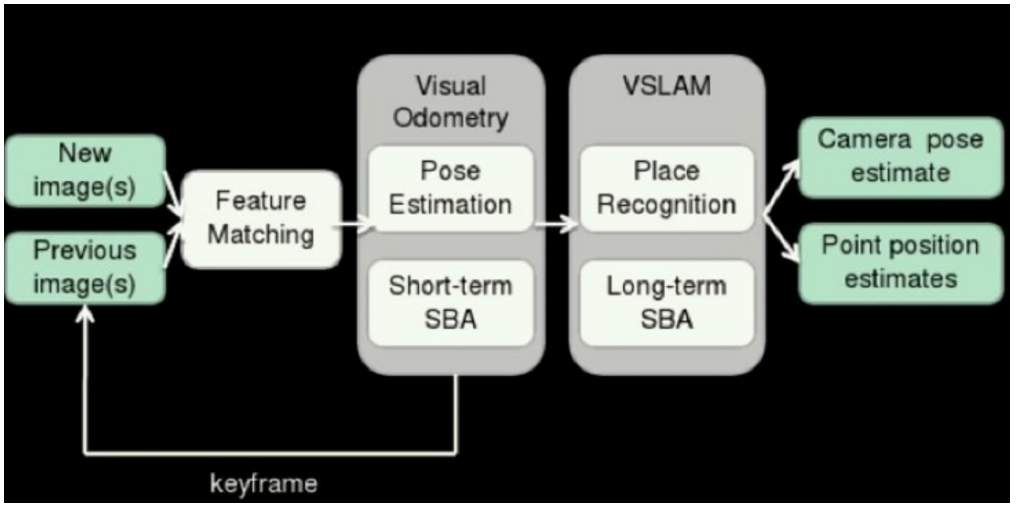
(1) A set of feature points is extracted from the images – ideally points such as corners which can be reliably identified in subsequent images as well.

(2) One determines a correspondence of these points across the various images. This can be done either through local tracking (using optical flow approaches) or by random sampling of possible partners based on a feature descriptor (SIFT, SURF, etc.) associated with each point.

(3) The camera motion is estimated based on a set of corresponding points. In many approaches this is done by a series of algorithms such as the eight-point algorithm or the five-point algorithm followed by bundle adjustment.

(4) For a given camera motion one can then compute for example a dense reconstruction using photometric stereo approaches

SLAM Brief Idea



How to build a map?

Compute relative pose and 3D point locations from two views

How to localize yourself?

Pose from known 3D map → pose from point correspondences

- Feature detectors
- Feature matching (RANSAC to reduce outliers)
- Case: Pose from point correspondences (geometric error)
- Case: Pose from direct intensity comparisons (photometric error)

L05: Problem Formulation

Consider a set of n possibly (nonlinear) equations in m unknowns$x = [x_1, ..., x_m]^\top$

$$r(x) = \begin{bmatrix} r_1(x) \\ \vdots \\ r_n(x) \end{bmatrix} = 0$$

It is often not possible to find an exact solution to this problem

→ approximate solution that minimizes the sum of squares of the residuals

$$\min_x f(x) = \min_x \frac{1}{2} \sum_{i=1}^{m} r_i^2(x)$$

→ find the x that minimizes the objective function f

If the equations r are linear then we can obtain use linear least squares to solve the problem

If the equations are non-linear then we have to linearize the problem. → L05

L05: Iterative Optimization

Solution of the non-linear Least-Squares Problems

$$\min_x f(x) = \min_x \frac{1}{2} \sum_{i=1}^{m} r_i^2(x)$$

Iterative optimization - **Gauss-Newton Method (GN) / Levenberg Marquardt (LM):**

1. Start at an initial estimate

2. For $k = 0,1,\ldots.$ Linearize the problem by using Taylor Series Expansion

$$\min_{\Delta x} f(x_k + \Delta x) = \min_{\Delta x} \frac{1}{2} \|r(x_k + \Delta x)\|^2$$

3. Solve the linearized problem by using a linear Least-Square Problem

$$\min_{\Delta x} l(\Delta x) = \min_{\Delta x} \frac{1}{2} \|r(x_k) + J_r \Delta x\|^2$$

4. Update the estimate, return to step 2.

GN normal equations: $J^\top J \Delta x = -J^\top r(x_k) \quad x_{k+1} = x_k + \alpha \Delta x, \quad k = 0, 1, \ldots$

LM normal equations: $(J^\top J + \lambda diag(J^\top J))\Delta x = -J^\top r$

Maximum a Posteriori (MAP) Estimation

"Translation" to a Maximum a posteriori estimation problem:

- Motion problem (estimate sequence of camera poses [R|T]) → unknown **state variables**
- Structure problem (estimate 3D points) → unkonwn **state variables**
- 2D image points/3Dpoints in camera coordinates → given **measurements**
- **State parameters:** camera poses, robot velocity, sensor biases, camera intrinsics, 3D Geometry, 3D world point coordinates, etc.
- **Measurements:** Observations/observed data: 2D point positions, (geometric) noise on point positions, pixel intensities, (photometric) noise on pixel intensities

We are interested in the **unknown state variables** X, **given the measurements** Z.

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

$$
\begin{aligned}
X^{\mathrm{MAP}} &= \underset{X}{\operatorname{argmax}}\, p(X \mid Z) \\
&= \underset{X}{\operatorname{argmax}}\, \frac{p(Z \mid X)\, p(X)}{p(Z)} \\
&= \underset{X}{\operatorname{argmax}}\, l(X;Z)\, p(X) \\
& \qquad\qquad l(X;Z) \propto p(Z \mid X)
\end{aligned}
$$

Nonlinear MAP inference for state estimation

Let X be the set of all **unknown state variables**, and Z be the set of all **measurements**.

We are interested in **estimating the unknown state variables** X, given the measurements Z.

Nonlinear MAP inference for state estimation

$$X^{\text{MAP}} = \underset{X}{\text{argmax}}\, p(X \mid Z)$$

A state variable x is typically used to describe the physical state of an object.

We can estimate several state variables at once by concatenating all the variables into the vector x:

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_p \end{bmatrix}$$

The equations $r_i(x)$ can be defined to operate on one or more of the p state variables.

MAP – Problem Formulation

Minimize error over the state variables $X = \{[R|T]_i\}\{\mathbf{x}_j\}$

with the measurements $Z = \{\mathbf{x}'_{ij}\}$

This gives us the measurement prediction function

$$\hat{\mathbf{z}}_i = h_i(X_i) = h_i([R|T]_i, \mathbf{x}_i) = \pi_i(g([R|T], \mathbf{x}))$$

Measurement likelihood:

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

MAP estimate:

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

Linearize the problem

Solve with linear least squares

Iterative method: Gauss Newton

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

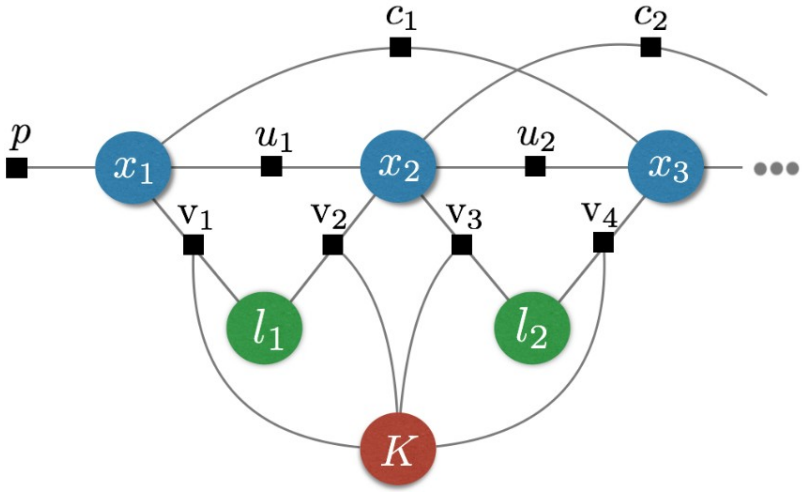


Fig. 3: **SLAM as a factor graph**: Blue circles denote robot poses at consecutive time steps $(x_1, x_2, \ldots)$, green circles denote landmark positions $(l_1, l_2, \ldots)$, red circle denotes the variable associated with the intrinsic calibration parameters (K). Factors are shows are black dots: the label "u" marks factors corresponding to odometry constraints, "v" marks factors corresponding to camera observations, "c" denotes loop closures, and "p" denotes prior factors.

Map Optimization, Sensor Fusion with Factor Graphs

Combining many different sensors in SLAM is a difficult and highly nonlinear problem

• Factor graphs provide powerful tools for expressing and solving nonlinear estimation problems

• It has become the current standard for the formulation of SLAM

- variables correspond to nodes in the factor graph.
- Terms $p(\mathbf{z}_i \mid X_i)$ and the prior $p(X)$ are factors (they encode probabilistic constraints over a subset of nodes).

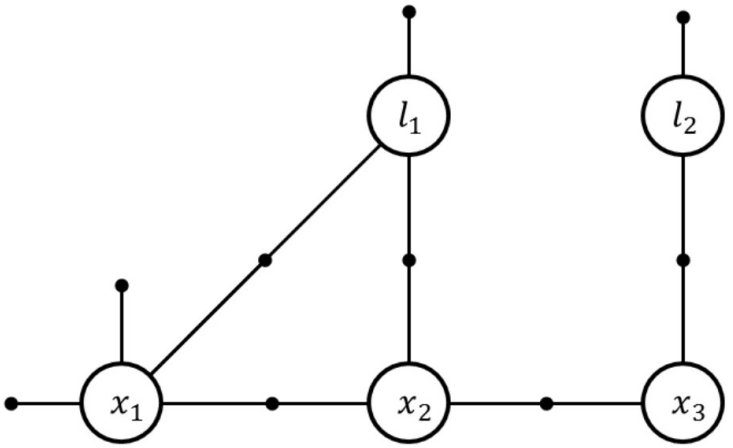
A **factor graph** is a graphical model that encodes

the dependence between the i-th factor (and its measurement

z_i) and the corresponding variables X_i .

Factor graph interpretation:

- enables an insightful vizualization
- generality: a factor graph can model complex inference problem with heterogeneous variables and factors, and arbitrary intercon



$$l(X;Z) \propto p(Z \mid X)$$

Optimization

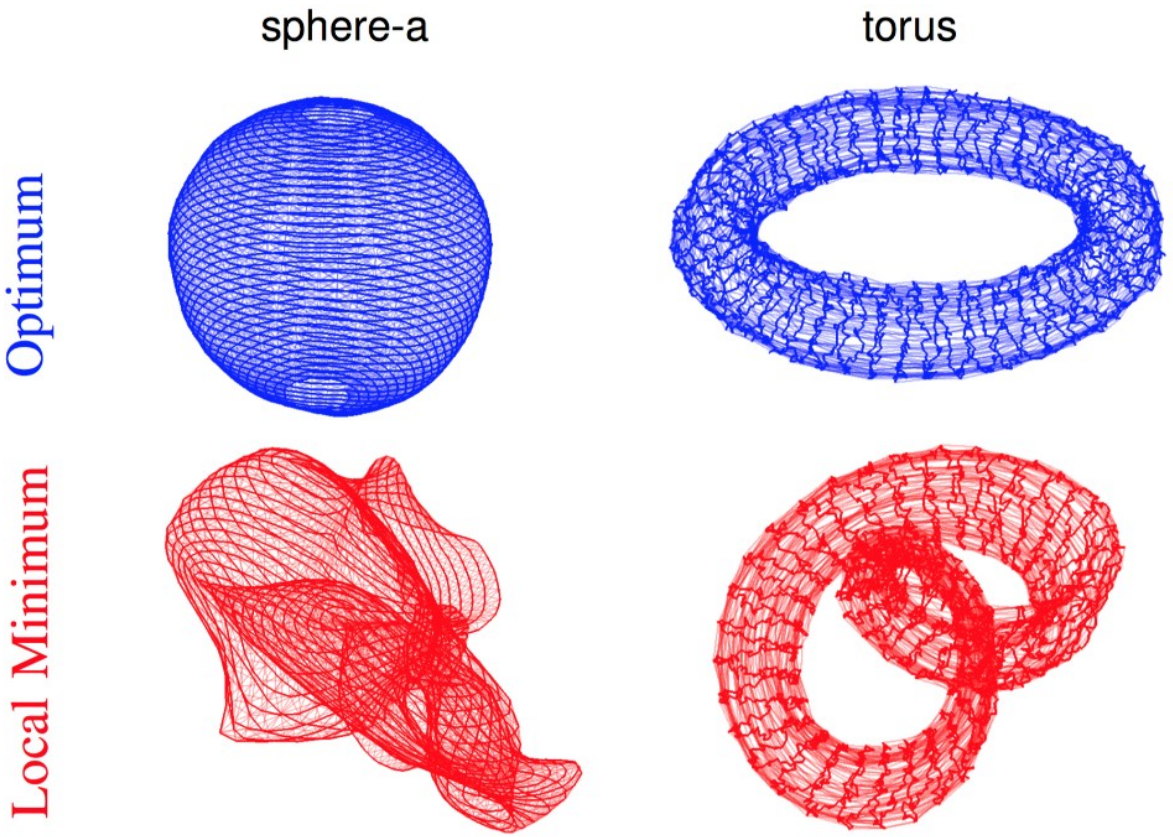


Fig. 7: The back-bone of most SLAM algorithms is the MAP estimation of robot trajectory, which is computed via non-convex optimization. A globally optimal solution corresponds to a correct map reconstruction. However, if the estimate does not converge to the global optimum, the map reconstruction is largely incorrect, hindering robot operation. Recent theoretical tools are enabling detection of wrong convergence episodes, and are opening avenues for fail detection and recovery techniques.

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 a SLAM system

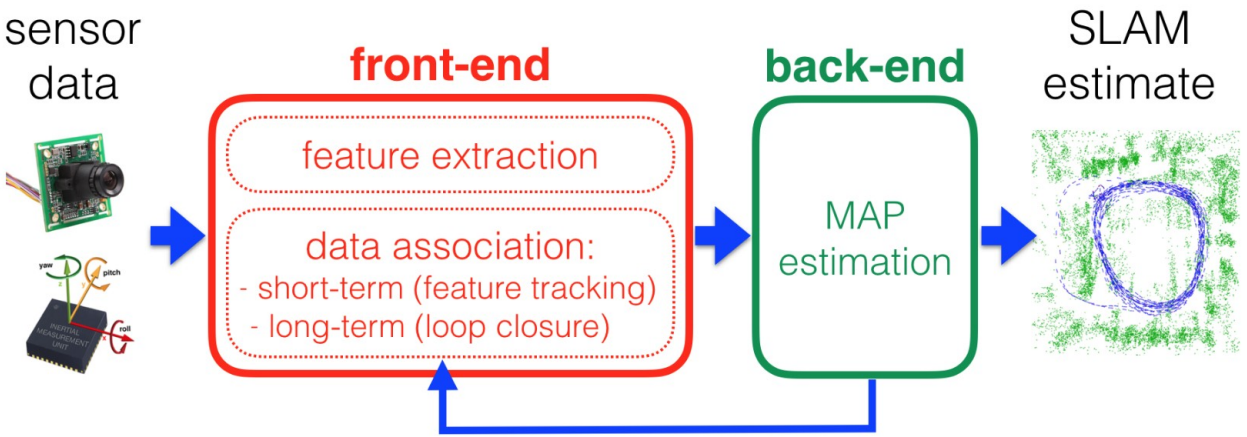


Fig. 2: Front-end and back-end in a typical SLAM system. The back-end can provide feedback to the front-end for loop closure detection and verification.

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

- Short-term tracking
  - Pose estimation given the map
  - Keyframe proposals
- Long-term tracking
  - Visual place recognition
  - Loop closure detection over keyframes

  (Lowry, S. et al. (2016). Visual Place Recognition: A Survey. IEEE Transactions on Robotics, 32(1), 1–19.)
- Mapping
  - Building and optimizing the map over keyframes
  - Data fusion

Appearance-based Place Recognition

How can we recognize that we have been visiting the same place before?

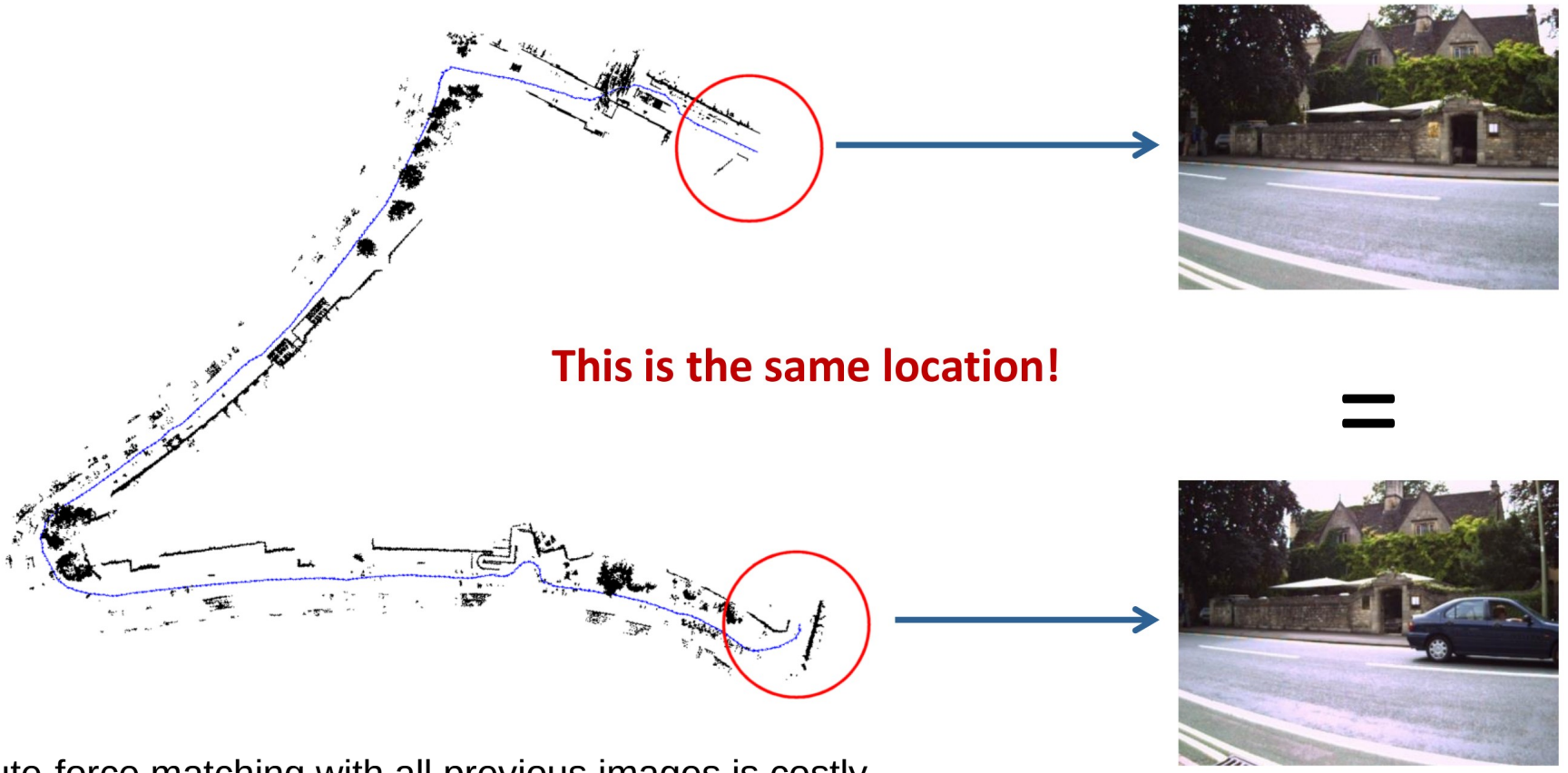
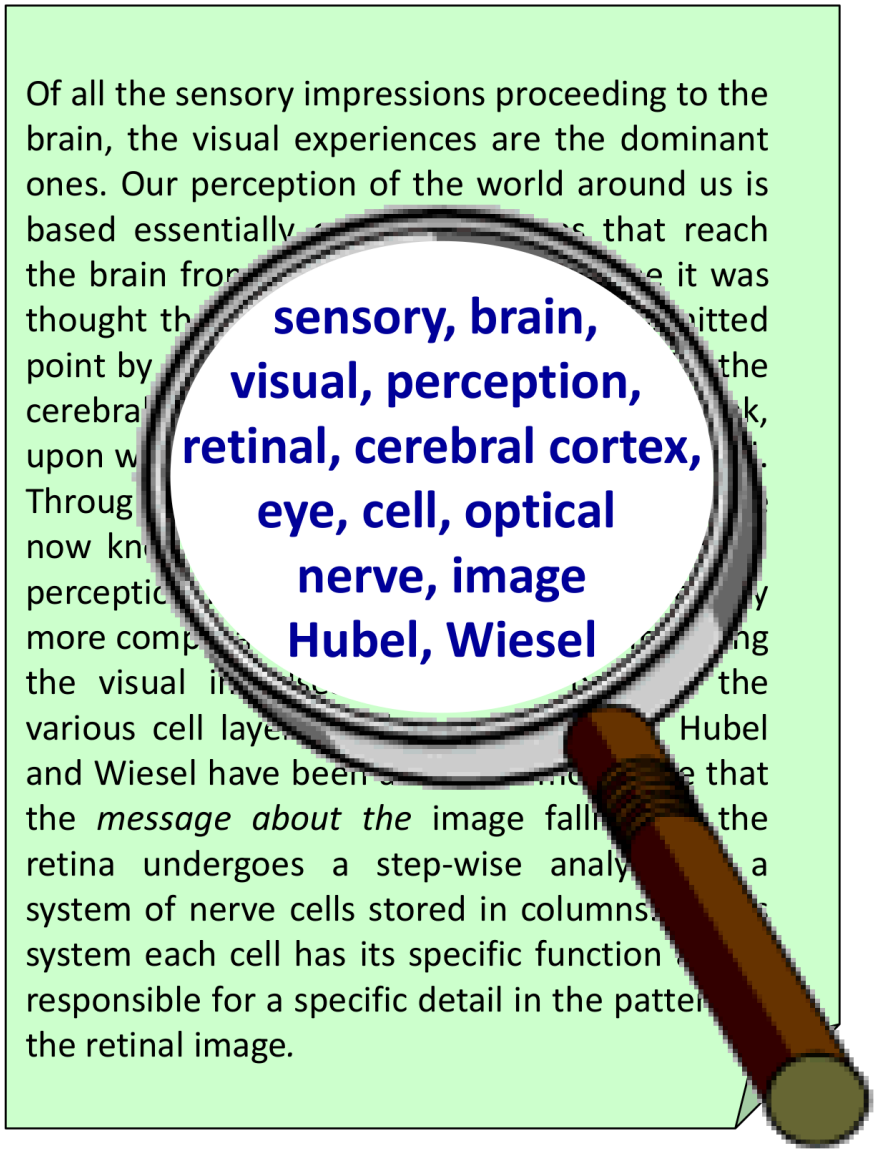


Image: J. Sturm (TUM)

Brute-force matching with all previous images is costly

How to make the search more efficient?

Analogy to Document Retrieval

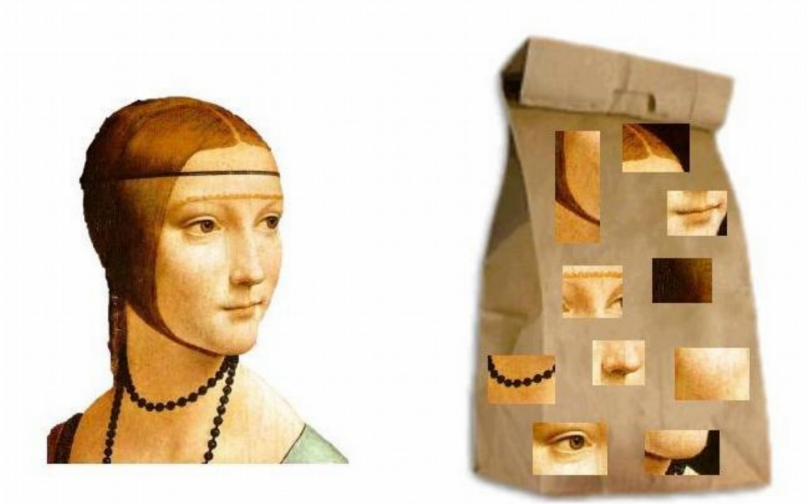


J. Sturm (TUM)

Object/Scene Recognition

Analogy to documents: The content can be inferred from the frequency of visual words

Visual words = (independent) features



Object

bag of visual words

Face

features

J. Sturm (TUM)

Bag of Visual Words

Visual words = (independent) features
Construct a dictionary of representative words

dictionary of visual words (codebook)

J. Sturm (TUM)

Bag of Visual Words

- Visual words = (independent) features
- Construct a dictionary of representative words
- Represent the image based on a histogram of word occurrences (bag)

Each detected feature is assigned to the closest entry in the codebook

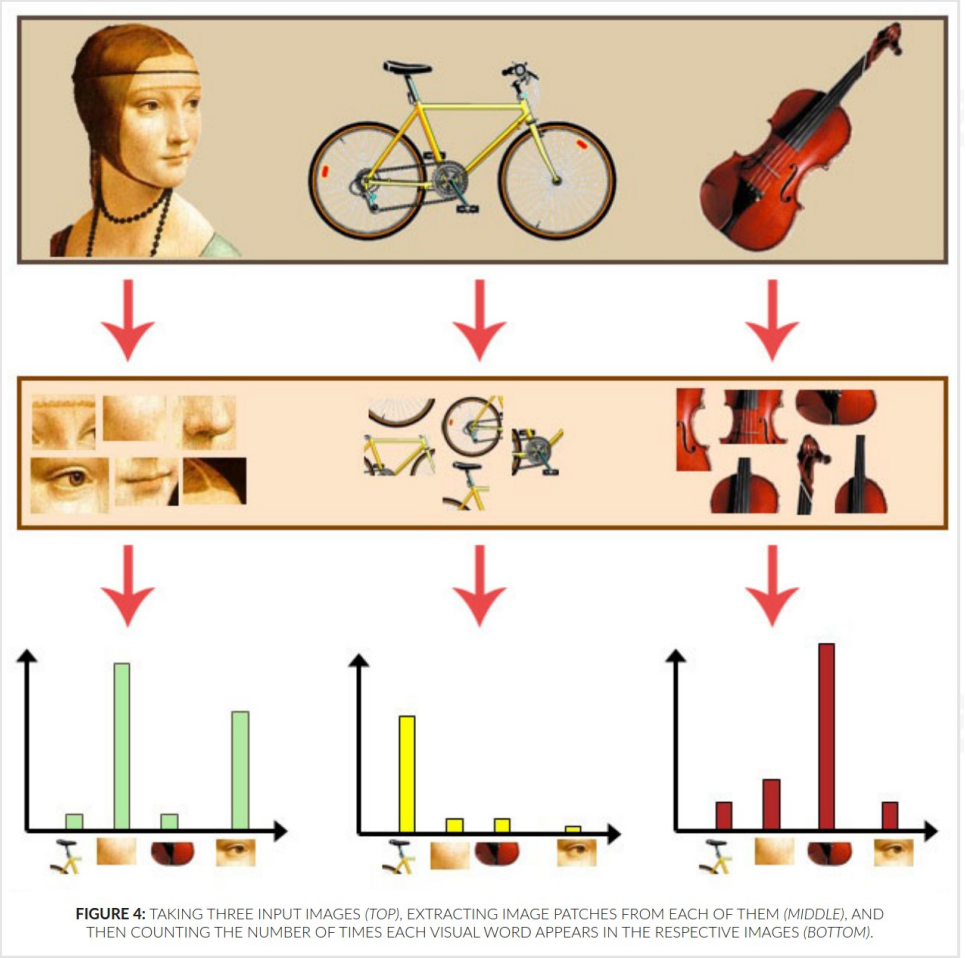
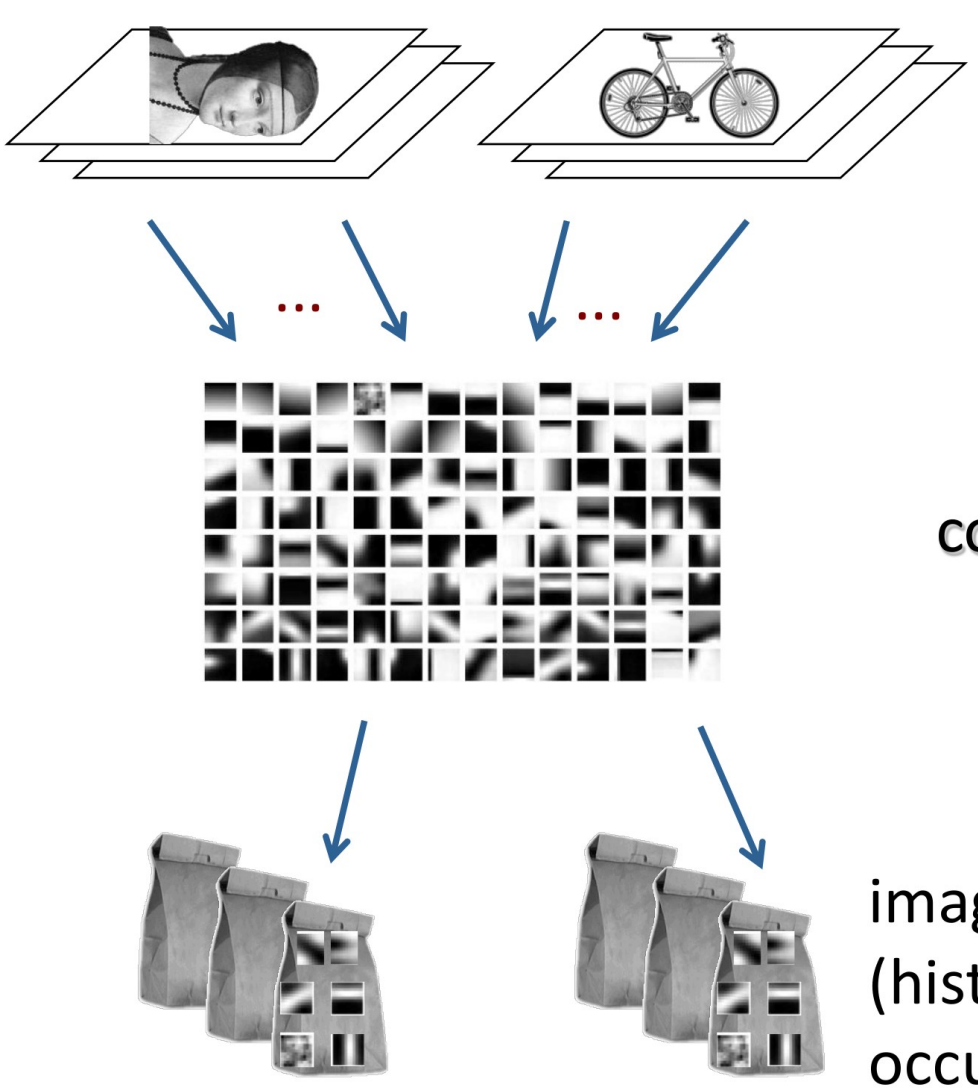


FIGURE 4: TAKING THREE INPUT IMAGES *(TOP)*, EXTRACTING IMAGE PATCHES FROM EACH OF THEM *(MIDDLE)*, AND THEN COUNTING THE NUMBER OF TIMES EACH VISUAL WORD APPEARS IN THE RESPECTIVE IMAGES *(BOTTOM)*.

J. Sturm (TUM)

Bag of Visual Words - Overview



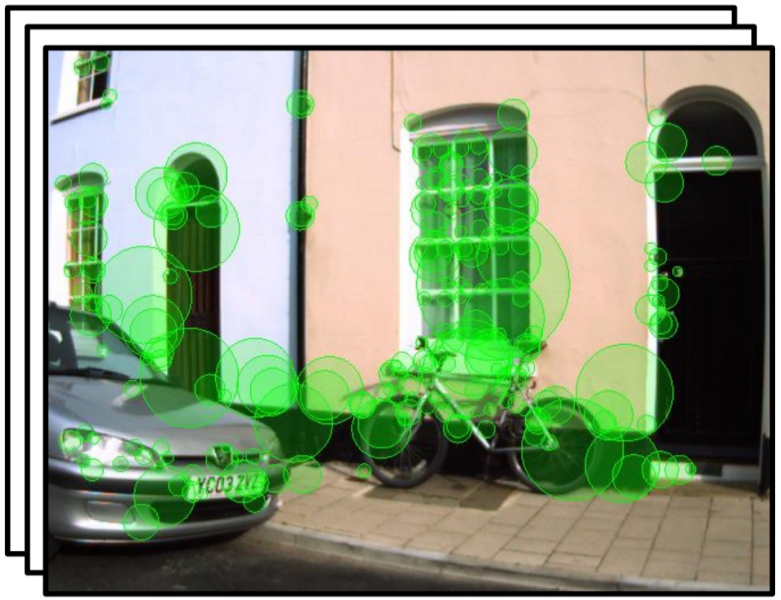
feature detection and extraction (e.g., SIFT, …)

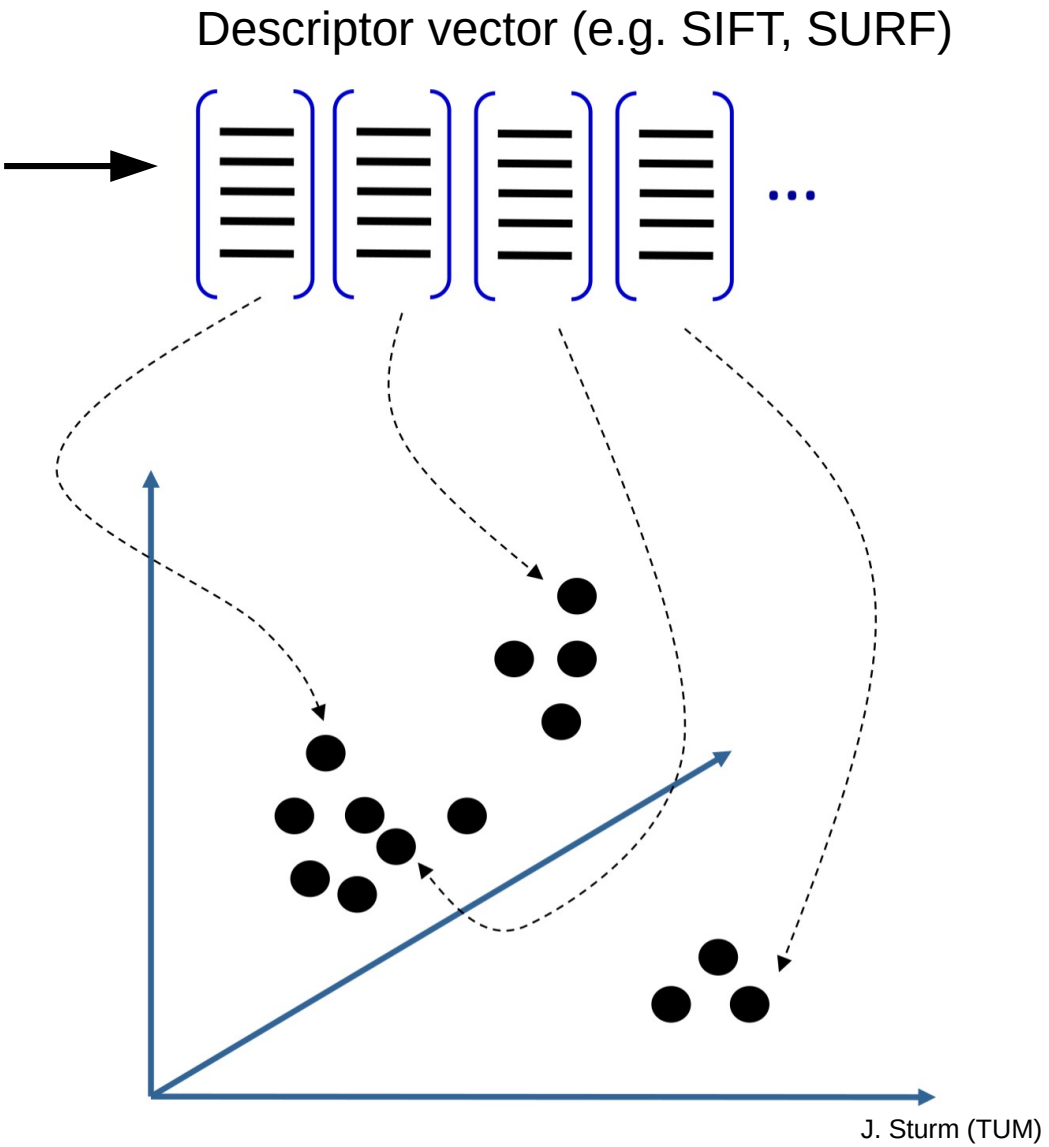
codewords dictionary

image representation (histogram of word occurrences)

J. Sturm (TUM)

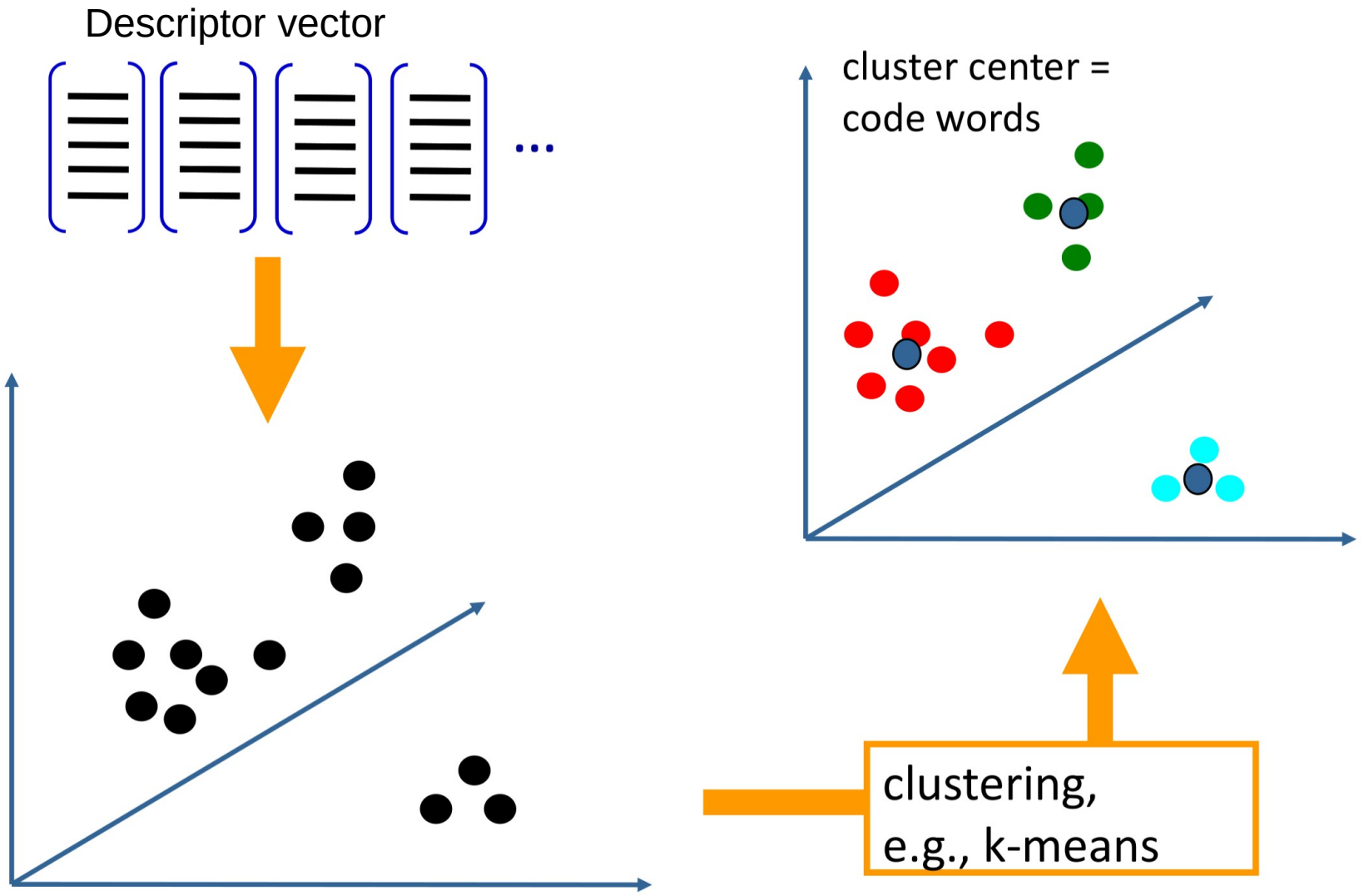
Leaning the Dictionary





J. Sturm (TUM)

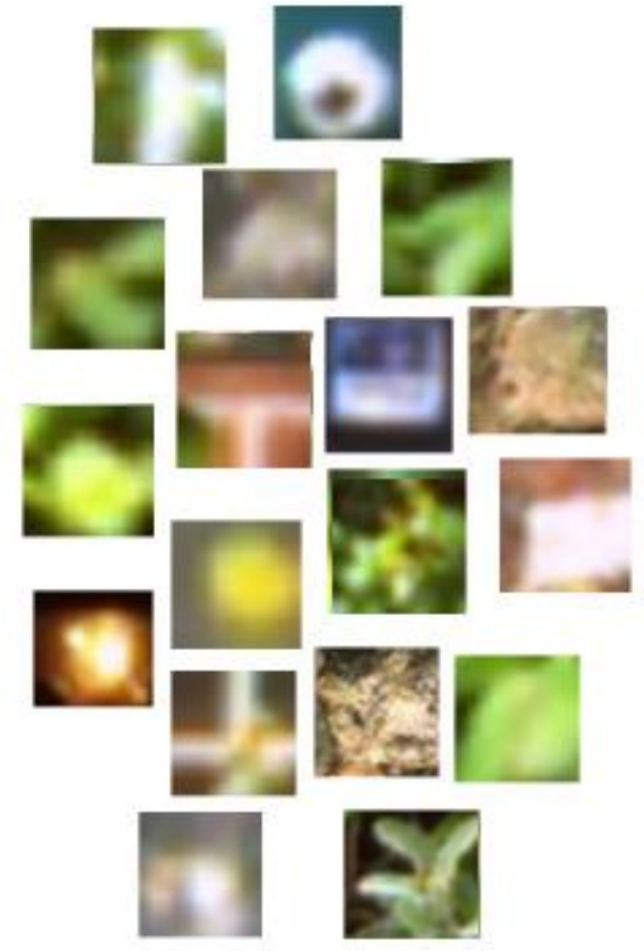
Leaning the Dictionary



J. Sturm (TUM)

Learning the Visual Vocabulary

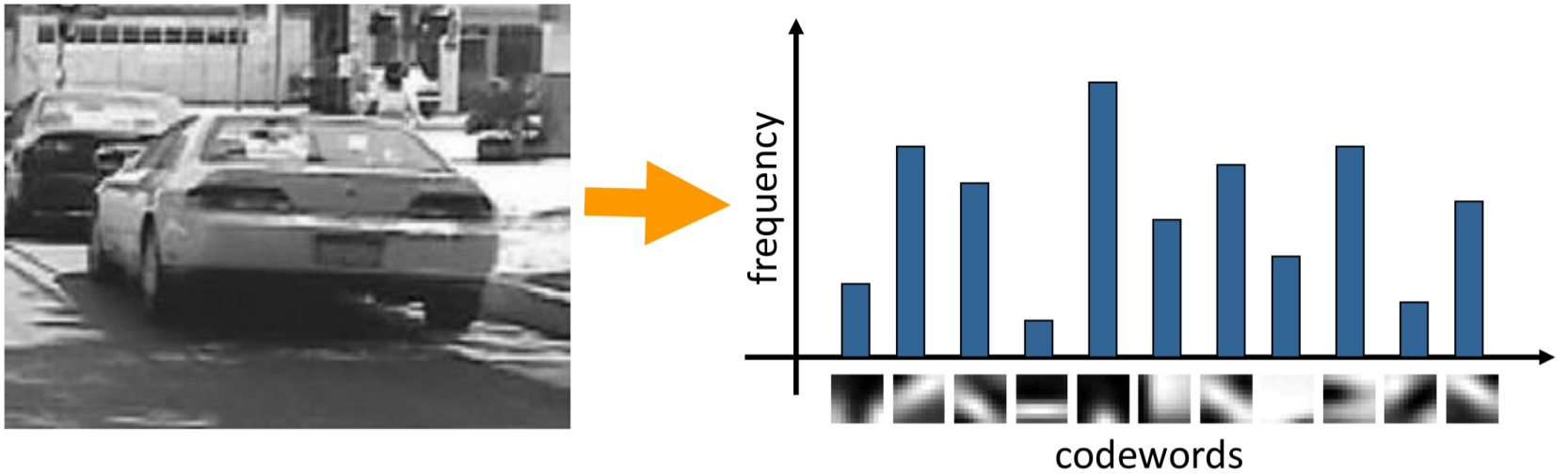
feature extraction & clustering



J. Sturm (TUM)

Example Image Representation

Build the histogram by assigning each detected feature to the closest entry in the codebook

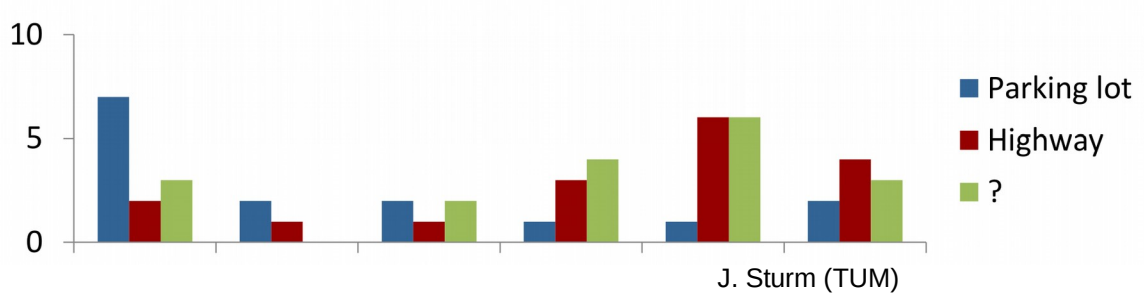


Compare histogram of new scene with those of known scenes, e.g., using

- simple histogram intersection

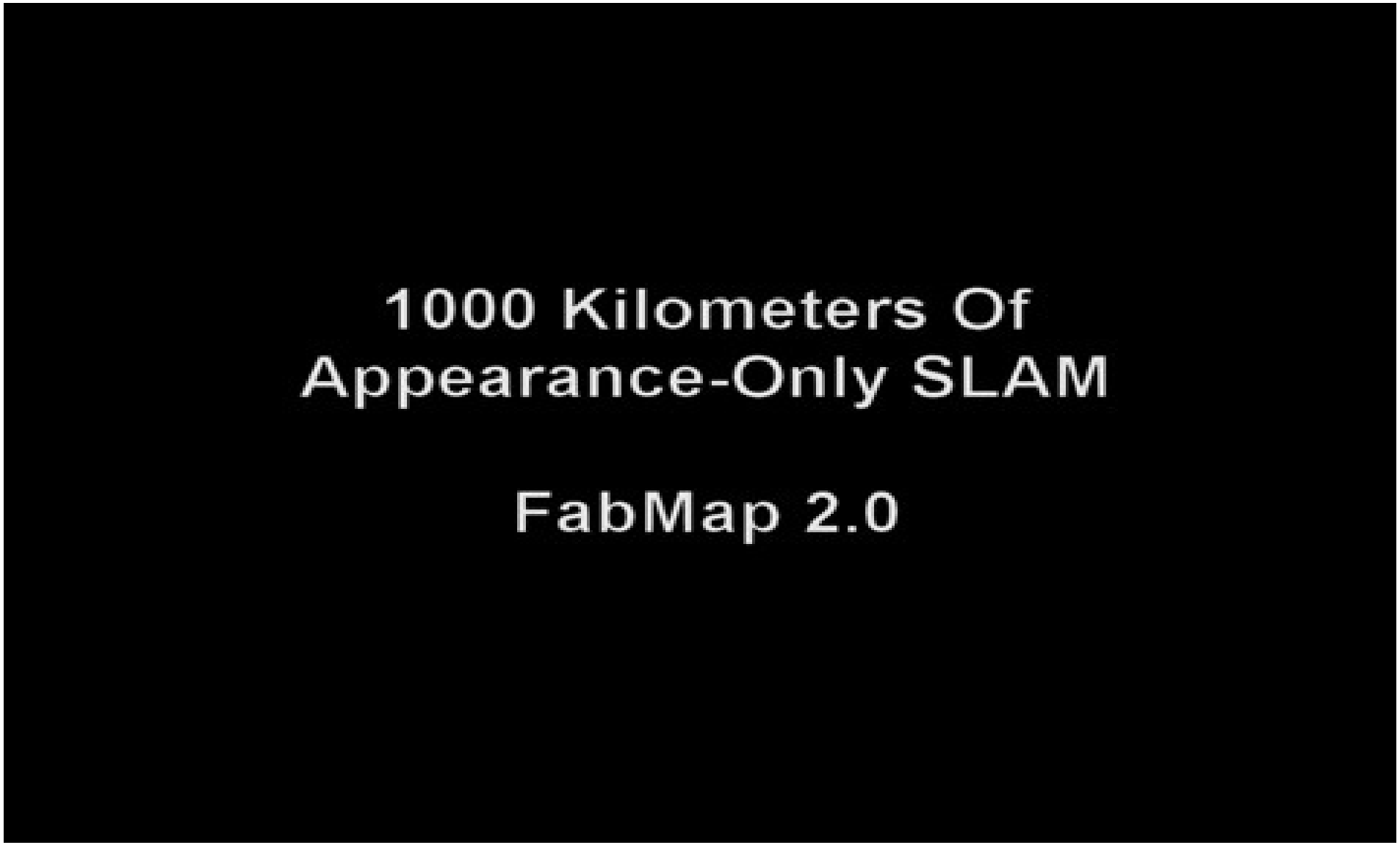
$$score(\mathbf{p}, \mathbf{q}) = \sum_i \min(p_i, q_i)$$

- naïve Bayes
- more advanced statistical methods



J. Sturm (TUM)

Example: FAB-MAP 2.0



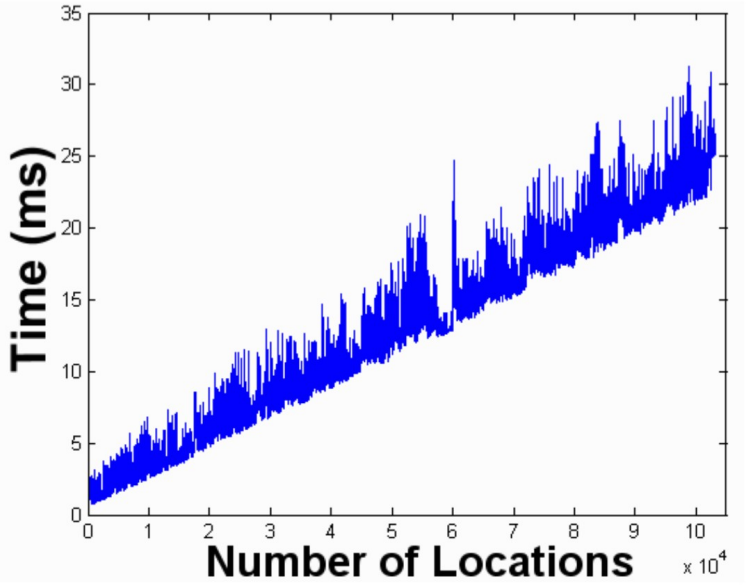
Cummins, M., & Newman, P. (2008). FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance. The International Journal of Robotics Research, 27(6), 647–665.

Bag of Visual Words

Time Performance

Inference: 25 ms for 100k locations

SURF detection + quantization: 483 ms



Bag of Visual Words:

- Compact representation of content
- Highly efficient and scalable
- Requires training of a dictionary
- Insensitive to viewpoint changes/image
- deformations (inherited from feature descriptor)

[Fei-Fei and Perona, 2005; Nister and Stewenius, 2006]

Components of a SLAM system

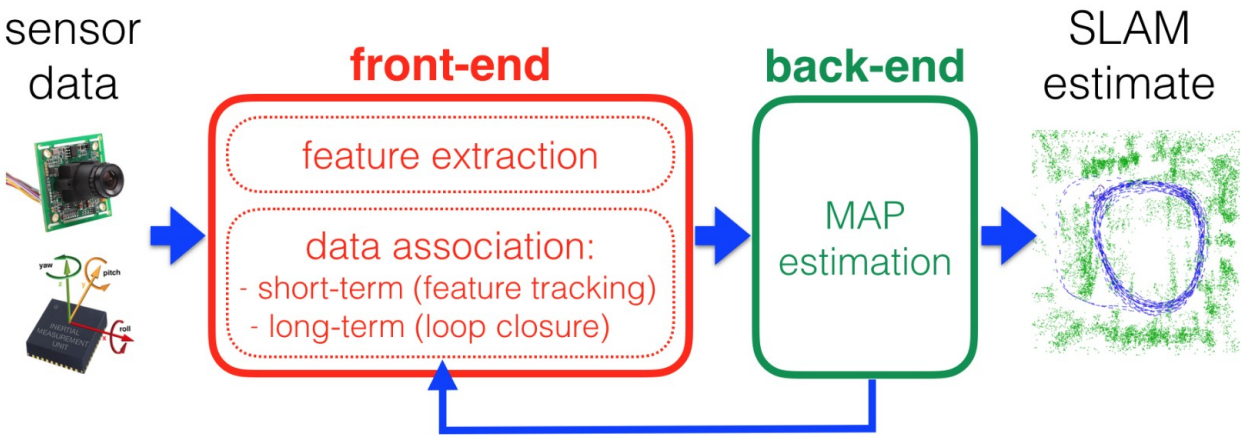


Fig. 2: Front-end and back-end in a typical SLAM system. The back-end can provide feedback to the front-end for loop closure detection and verification.

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

- Short-term tracking
  - Pose estimation given the map
  - Keyframe proposals

- Long-term tracking
  - Visual place recognition
  - Loop closure detection over keyframes

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

- Mapping
  - Building and optimizing the map over keyframes
  - Data fusion

LSD-SLAM: LARGE SCALE DIRECT SLAM

Content and illustrations from

Engel, J., Schöps, T., & Cremers, D. (2014), «LSD-SLAM: Large-Scale Direct Monocular SLAM»,

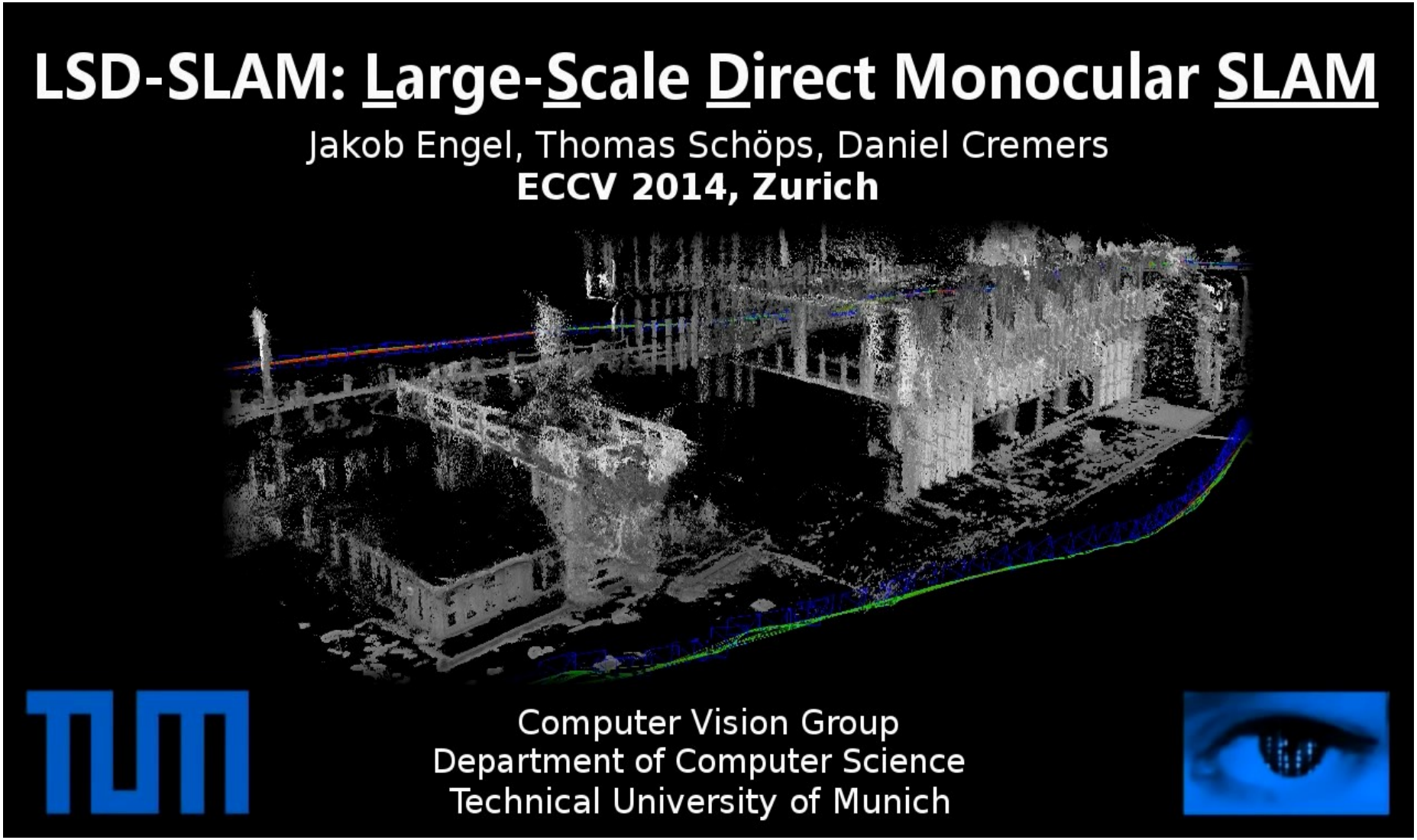
European Conference on Computer Vision (ECCV) (Vol. 8690)

Code: https://github.com/tum-vision/lsd_slam

Main contributions

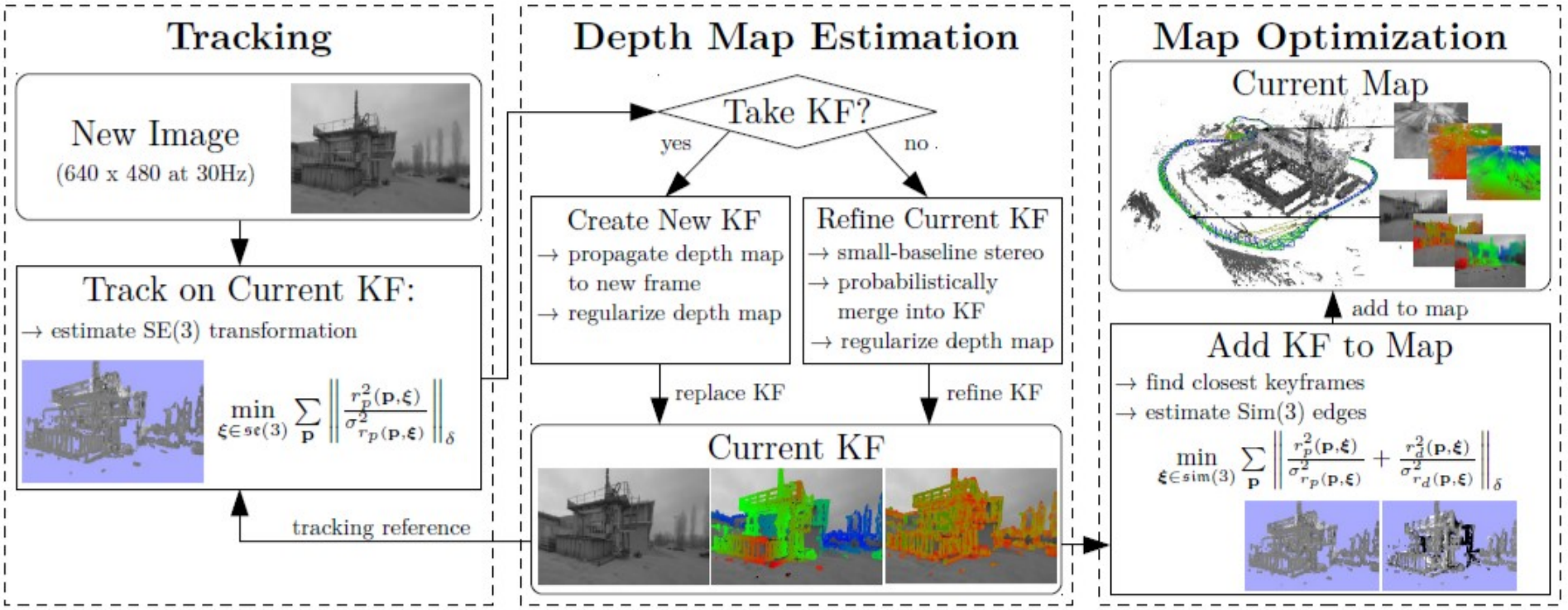
- A framework for large-scale, direct monocular SLAM
- Novel scale-aware direct image alignment algorithm
- Probabilistically consistent incorporation of uncertainty of the estimated depth into tracking

LSD-SLAM


LSD-SLAM: Large-Scale Direct Monocular SLAM
Jakob Engel, Thomas Schöps, Daniel Cremers
ECCV 2014, Zurich
Computer Vision Group
Department of Computer Science
Technical University of Munich

LSD-SLAM System Overview

The algorithm consists of three major components:



Engel, J., Schöps, T., & Cremers, D. (2014).
«LSD-SLAM: Large-Scale Direct Monocular SLAM»,
European Conference on Computer Vision (ECCV) (Vol. 8690)

LSD-SLAM Short-term Tracking

- **Direct** tracking
- **Semi-dense:**

  Reduction to image-regions

  which carry information
- **Incorporation of**

  depth uncertainty

  into keyframe-based tracking

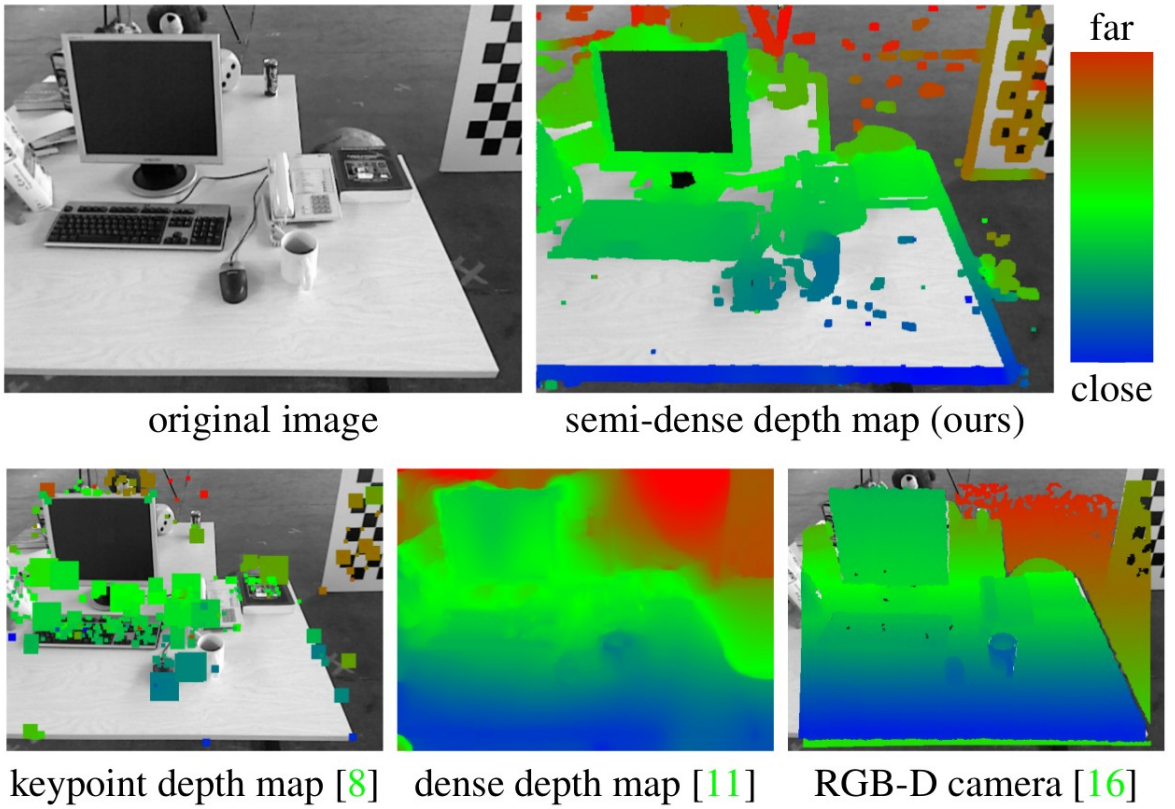


Figure 2. Semi-Dense Approach: Our approach reconstructs and tracks on a *semi-dense inverse depth map*, which is dense in all image regions carrying information (top-right). For comparison, the bottom row shows the respective result from a keypoint-based approach, a fully dense approach and the ground truth from an RGB-D camera.

Engel, J., Sturm, J., & Cremers, D. (2013). Semi-dense Visual Odometry for a Monocular Camera. In 2013 IEEE International Conference on Computer Vision

LSD-SLAM Short-term Tracking

From an existing keyframe $\mathcal{K}_i = (I_i, D_i, V_i)$

estimate the relative pose $\xi_{ji} \in se(3)$ of a new image I_j by minimizing

the variance-normalized photometric error:

$$E_p(\boldsymbol{\xi}_{ji}) = \sum_{\mathbf{p} \in \Omega_{D_i}} \left\| \frac{r_p^2(\mathbf{p}, \boldsymbol{\xi}_{ji})}{\sigma^2_{r_p(\mathbf{p}, \boldsymbol{\xi}_{ji})}} \right\|_\delta$$

$$\text{with} \quad r_p(\mathbf{p}, \boldsymbol{\xi}_{ji}) := I_i(\mathbf{p}) - I_j(\omega(\mathbf{p}, D_i(\mathbf{p}), \boldsymbol{\xi}_{ji}))$$

$$\sigma^2_{r_p(\mathbf{p}, \boldsymbol{\xi}_{ji})} := 2\sigma_I^2 + \left(\frac{\partial r_p(\mathbf{p}, \boldsymbol{\xi}_{ji})}{\partial D_i(\mathbf{p})} \right)^2 V_i(\mathbf{p})$$

$$\text{where } \| \cdot \|_\delta \text{ is the Huber norm}$$

$$\|r^2\|_\delta := \begin{cases} \frac{r^2}{2\delta} & \text{if } |r| \leq \delta \\ |r| - \frac{\delta}{2} & \text{otherwise.} \end{cases}$$

Engel, J., Sturm, J., & Cremers, D. (2013). Semi-dense Visual Odometry for a Monocular Camera. In 2013 IEEE International Conference on Computer Vision

Estimating Uncertainty in Inverse Depth

The optimal inverse depth d^* as a function of the noisy inputs, and its variance:

$$d^* = d(I_0, I_1, \xi, \pi) \qquad \sigma_d^2 = J_d \Sigma J_d^\top$$

Split this computation into three steps:

1. Computation of the epipolar line
   - Geometric error from noise on ξ and π
2. Search for correct disparity
   - Photometric error from noise on I_0 and I_1
3. Inverse depth from disparity
   - Scales the previous errors by a factor depending on the baseline

Engel, J., Sturm, J., & Cremers, D. (2013). Semi-dense Visual Odometry for a Monocular Camera. In 2013 IEEE International Conference on Computer Vision (pp. 1449–1456)

Estimating Uncertainty in Inverse Depth

Geometric disparity error:

Engel, J., Sturm, J., & Cremers, D. (2013). Semi-dense Visual Odometry for a Monocular Camera. In 2013 IEEE International Conference on Computer Vision (pp. 1449– 1456)

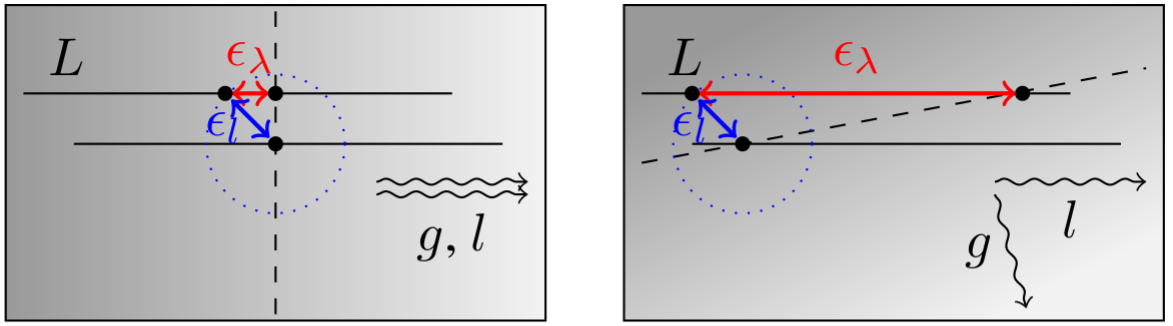


Figure 5. Geometric Disparity Error: Influence of a small positioning error ϵ_l of the epipolar line on the disparity error ϵ_λ. The dashed line represents the isocurve on which the matching point has to lie. ϵ_λ is small if the epipolar line is parallel to the image gradient (left), and a large otherwise (right).

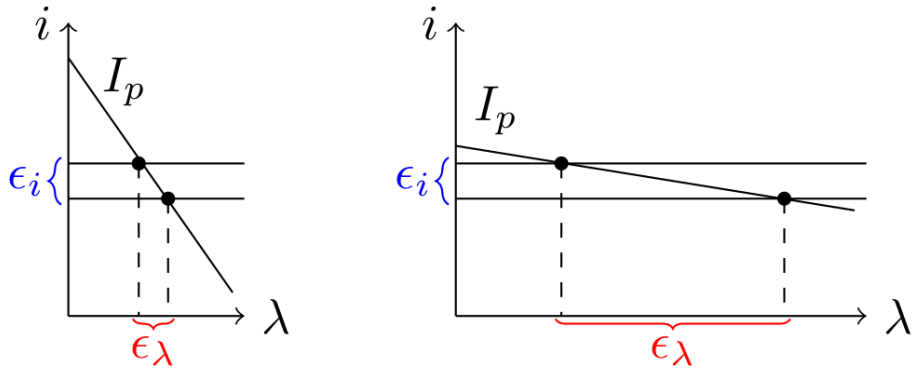
$$l_0 + \lambda^* \begin{pmatrix} l_x \\ l_y \end{pmatrix} \stackrel{!}{=} g_0 + \gamma \begin{pmatrix} -g_y \\ g_x \end{pmatrix}, \qquad \gamma \in \mathbb{R}$$

$$\lambda^*(l_0) = \frac{\langle g, g_0 - l_0 \rangle}{\langle g, l \rangle}$$

$$\sigma^2_{\lambda(\xi,\pi)} = J_{\lambda^*(l_0)} \begin{pmatrix} \sigma_l^2 & 0 \\ 0 & \sigma_l^2 \end{pmatrix} J^T_{\lambda^*(l_0)} = \frac{\sigma_l^2}{\langle g, l \rangle^2},$$

Estimating Uncertainty in Inverse Depth

Photometric error:



Engel, J., Sturm, J., & Cremers, D. (2013). Semi-dense Visual Odometry for a Monocular Camera. In 2013 IEEE International Conference on Computer Vision (pp. 1449– 1456)

Figure 6. Photometric Disparity Error: Noise ϵ_i on the image intensity values causes a small disparity error ϵ_λ if the image gradient along the epipolar line is large (left). If the gradient is small, the disparity error is magnified (right).

$$\lambda^* = \min_{\lambda} \left(i_{\text{ref}} - I_p(\lambda)\right)^2,$$

$$\lambda^*(I) = \lambda_0 + \left(i_{\text{ref}} - I_p(\lambda_0)\right) g_p^{-1},$$

$$\sigma^2_{\lambda(I)} = J_{\lambda^*(I)} \begin{pmatrix} \sigma_i^2 & 0 \\ 0 & \sigma_i^2 \end{pmatrix} J_{\lambda^*(I)} = \frac{2\sigma_i^2}{g_p^2},$$

Estimating Uncertainty in Inverse Depth

Pixel to inverse depth conversion:

$$\sigma_{d,\text{obs}}^2 = \alpha^2 \left(\sigma_{\lambda(\xi,\pi)}^2 + \sigma_{\lambda(I)}^2 \right), \qquad \alpha := \frac{\delta_d}{\delta_\lambda},$$

where δ_d denotes the length of the inverse depth interval and δ_λ the length of the epipolar line segment.

Inverse depth observation fusion (prior and observed):

$$\mathcal{N}\left(\frac{\sigma_p^2 d_o + \sigma_o^2 d_p}{\sigma_p^2 + \sigma_o^2}, \frac{\sigma_p^2 \sigma_o^2}{\sigma_p^2 + \sigma_o^2} \right)$$

Engel, J., Sturm, J., & Cremers, D. (2013). Semi-dense Visual Odometry for a Monocular Camera. In 2013 IEEE International Conference on Computer Vision (pp. 1449– 1456)

Estimating Uncertainty in Inverse Depth

In summary:

The stereo observation accuracy is approximated with three factors:

- The **photometric disparity error** $\sigma^2_{\lambda(I)}$,

  depending on the magnitude of the image gradient along the epipolar line
- The **geometric disparity error** $\sigma^2_{\lambda(\xi,\pi)}$,

  depending on the angle between the image gradient and the epipolar line
- The **pixel to inverse depth ratio** α ,

  depending on the camera translation, the focal length and the pixel's position

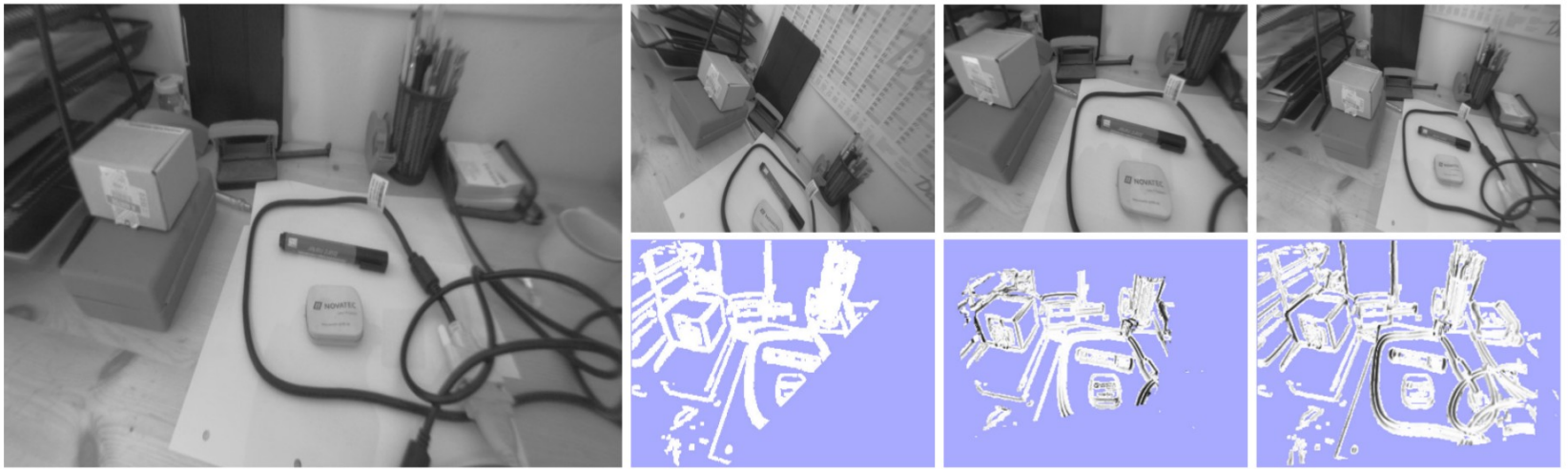
$$\sigma^2_{d,\text{obs}} = \alpha^2 \left(\sigma^2_{\lambda(\xi,\pi)} + \sigma^2_{\lambda(I)} \right),$$

Engel, J., Sturm, J., & Cremers, D. (2013). Semi-dense Visual Odometry for a Monocular Camera. In 2013 IEEE International Conference on Computer Vision (pp. 1449– 1456)

Estimating Uncertainty in Tracking

Inverse variance $\sigma_{r_p}^{-2}$ of the residual:

$$\sigma^2_{r_p(\mathbf{p},\boldsymbol{\xi}_{ji})} := 2\sigma_I^2 + \left(\frac{\partial r_p(\mathbf{p},\boldsymbol{\xi}_{ji})}{\partial D_i(\mathbf{p})}\right)^2 V_i(\mathbf{p})$$



(a) reference image (b) rotation (c) z trans. (d) x trans.

Fig. 4: Statistic normalization: (a) reference image. (b-d): tracked images and inverse variance $\sigma_{r_p}^{-2}$ of the residual. For pure rotation, depth noise has no effect on the residual noise and hence all normalization factors are the same. For z translation depth noise has no effect for pixels in the center of the image, while for x translation it only affects residuals with intensity-gradient in x direction.

Engel, J., Sturm, J., & Cremers, D. (2013). Semi-dense Visual Odometry for a Monocular Camera. In 2013 IEEE International Conference on Computer Vision (pp. 1449– 1456)

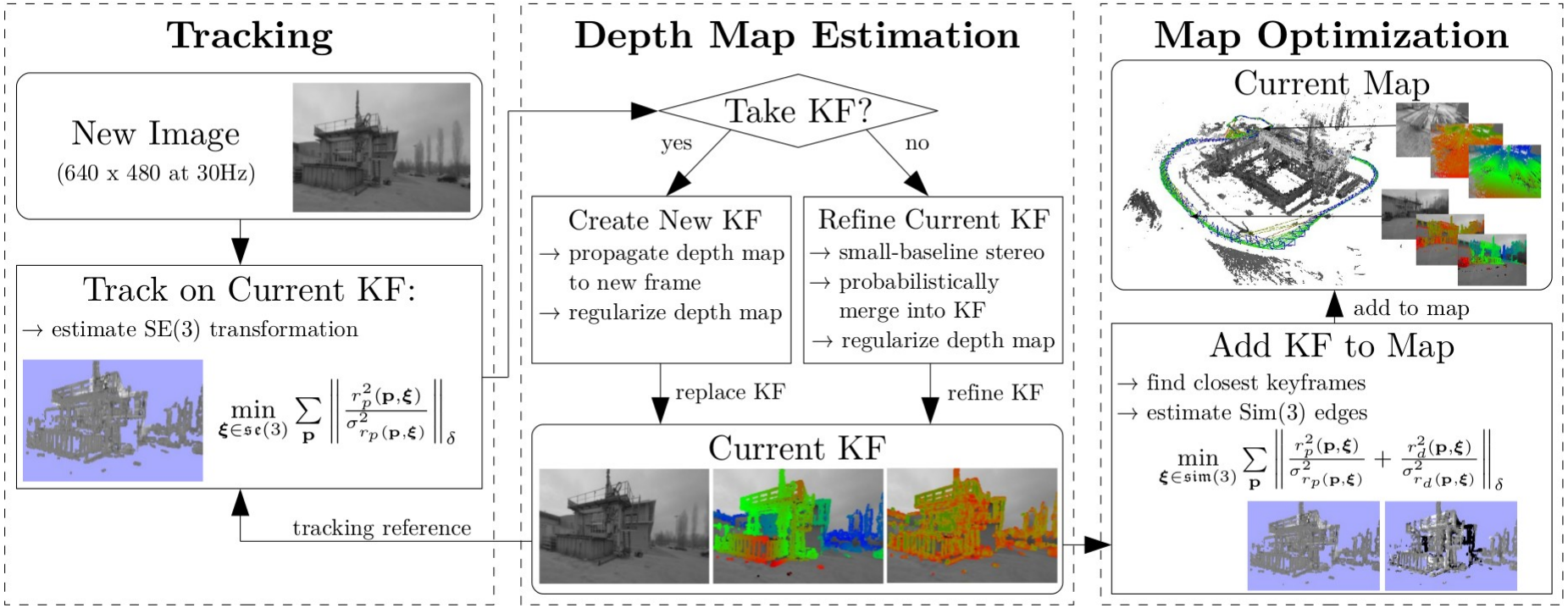
Keyframe

When the camera has moved too far,

a new keyframe is initialized by projecting points from existing,

close-by keyframes into it.

Mapping



Keyframe insertion

- When a keyframe is replaced as tracking reference, it is incorporated into the global map
- Long-term tracking is performed to find loop closures

Long-term Tracking

Keyframe insertion

- Long-term tracking is performed on the last keyframe in the mapping thread
- Loop closure candidates:
  - Closest 10 keyframes
  - Candidate from FAB-MAP
- Direct alignment between candidates
- Reciprocal tracking check

Long-term Tracking

Scale-drift aware direct image alignment

Between keyframes $i,\ j,$ estimate the relative pose ξ_{ij} by minimizing the error function:

$$E(\boldsymbol{\xi}_{ji}) := \sum_{\mathbf{p}\in\Omega_{D_i}} \left\| \frac{r_p^2(\mathbf{p},\boldsymbol{\xi}_{ji})}{\sigma^2_{r_p(\mathbf{p},\boldsymbol{\xi}_{ji})}} + \frac{r_d^2(\mathbf{p},\boldsymbol{\xi}_{ji})}{\sigma^2_{r_d(\mathbf{p},\boldsymbol{\xi}_{ji})}} \right\|_\delta ,$$

where the photometric residual is computed as

$$r_p(\mathbf{p},\boldsymbol{\xi}_{ji}) := I_i(\mathbf{p}) - I_j(\omega(\mathbf{p}, D_i(\mathbf{p}), \boldsymbol{\xi}_{ji}))$$

$$\sigma^2_{r_p(\mathbf{p},\boldsymbol{\xi}_{ji})} := 2\sigma_I^2 + \left(\frac{\partial r_p(\mathbf{p},\boldsymbol{\xi}_{ji})}{\partial D_i(\mathbf{p})}\right)^2 V_i(\mathbf{p})$$

The depth residual and its variance is computed as

$$r_d(\mathbf{p},\boldsymbol{\xi}_{ji}) := [\mathbf{p}']_3 - D_j([\mathbf{p}']_{1,2})$$

$$\sigma^2_{r_d(\mathbf{p},\boldsymbol{\xi}_{ji})} := V_j([\mathbf{p}']_{1,2}) \left(\frac{\partial r_d(\mathbf{p},\boldsymbol{\xi}_{ji})}{\partial D_j([\mathbf{p}']_{1,2})}\right)^2 + V_i(\mathbf{p}) \left(\frac{\partial r_d(\mathbf{p},\boldsymbol{\xi}_{ji})}{\partial D_i(\mathbf{p})}\right)^2$$

$$\mathbf{p}' := \omega_s(\mathbf{p}, D_i(\mathbf{p}), \boldsymbol{\xi}_{ji})$$

Engel, J., Sturm, J., & Cremers, D. (2013). Semi-dense Visual Odometry for a Monocular Camera. In 2013 IEEE International Conference on Computer Vision (pp. 1449– 1456)

Long-term Tracking

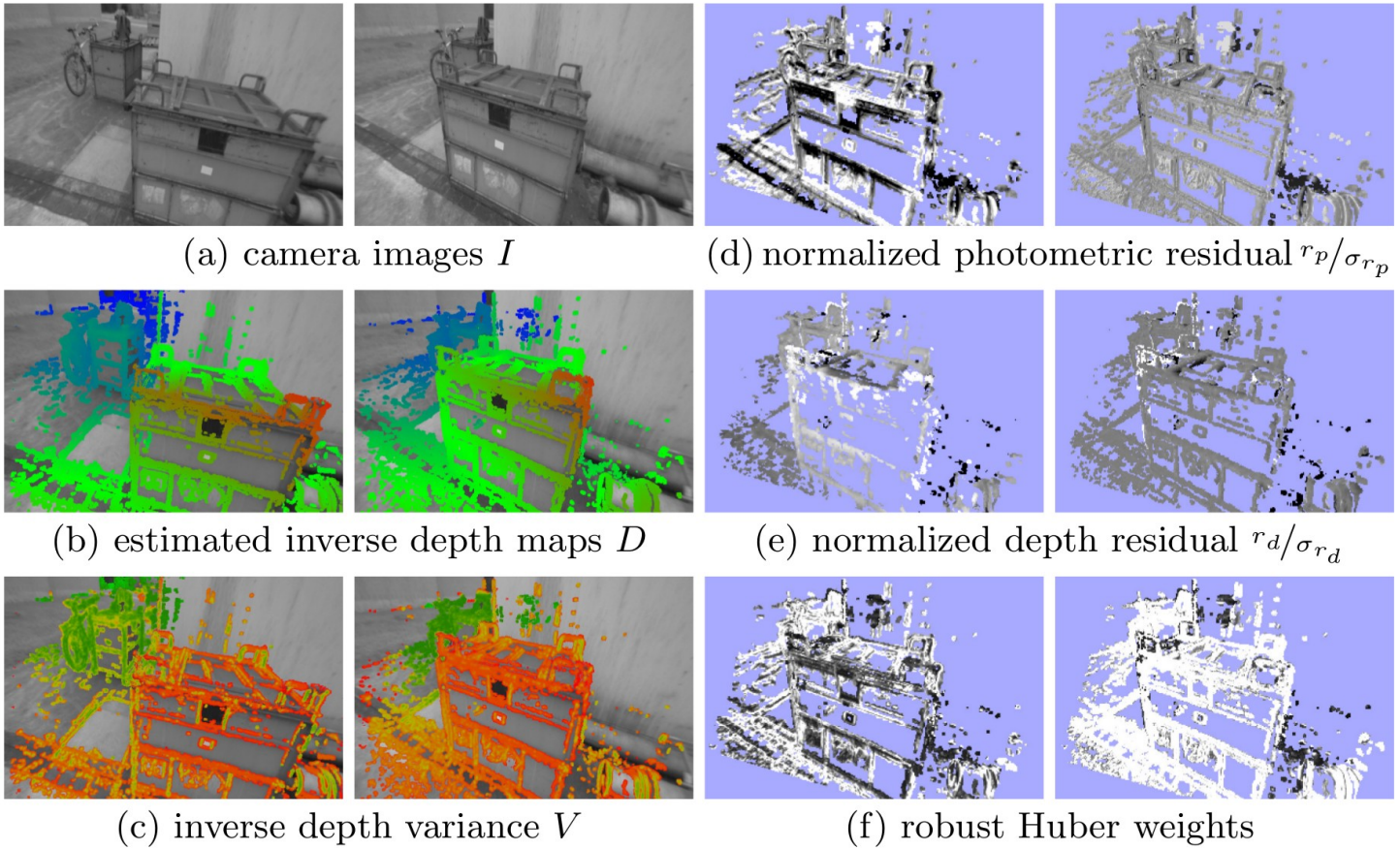


Fig. 5: Direct keyframe alignment on $\mathfrak{sim}(3)$: (a)-(c): two keyframes with associated depth and depth variance. (d)-(f): photometric residual, depth residual and Huber weights, before minimization (left), and after minimization (right).

Engel, J., Sturm, J., & Cremers, D. (2013). Semi-dense Visual Odometry for a Monocular Camera. In 2013 IEEE International Conference on Computer Vision (pp. 1449– 1456)

Long-term Tracking

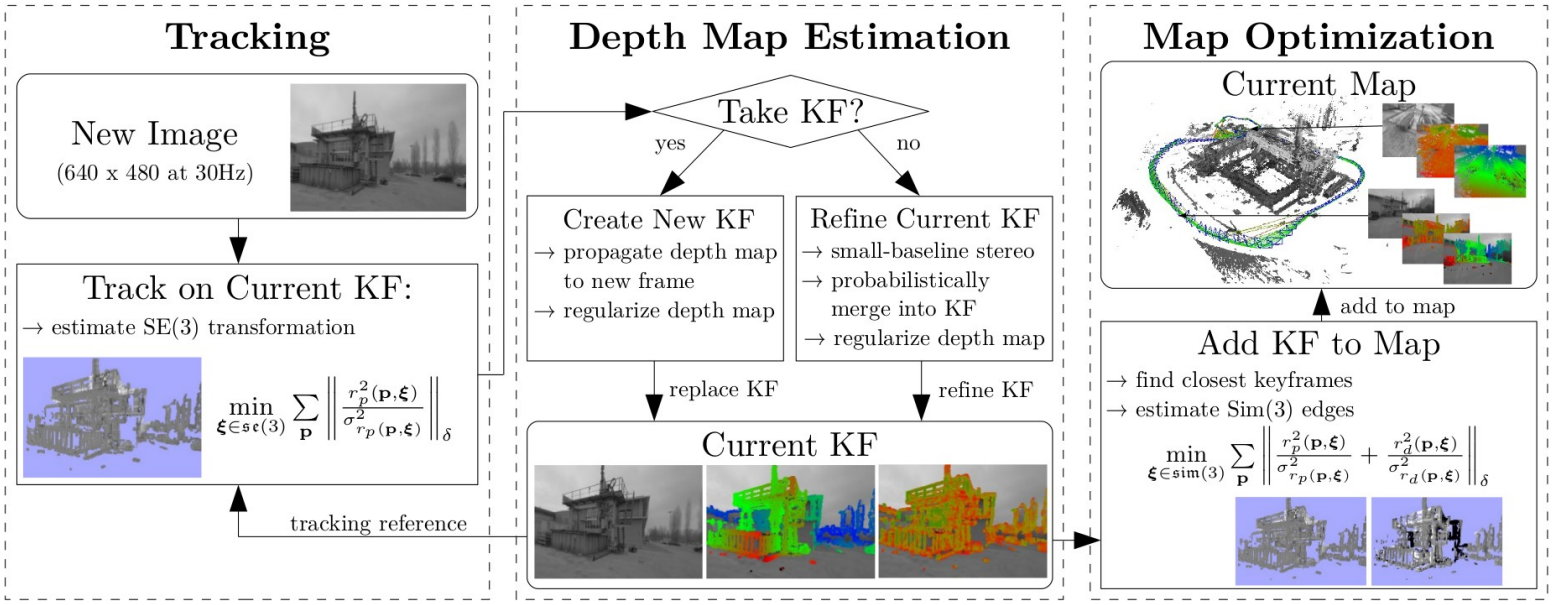
Reciprocal tracking check on loop closures:

For each candidate $\mathcal{K}_{j_k}$ we independently track $\boldsymbol{\xi}_{j_k i}$ and $\boldsymbol{\xi}_{i j_k}$

Only if the two estimates are statistically similar, i.e., if

$$e(\boldsymbol{\xi}_{j_k i}, \boldsymbol{\xi}_{i j_k}) := (\boldsymbol{\xi}_{j_k i} \circ \boldsymbol{\xi}_{i j_k})^T \left(\boldsymbol{\Sigma}_{j_k i} + \mathrm{Adj}_{j_k i} \boldsymbol{\Sigma}_{i j_k} \mathrm{Adj}_{j_k i}^T \right)^{-1} (\boldsymbol{\xi}_{j_k i} \circ \boldsymbol{\xi}_{i j_k})$$

is sufficiently small, they are added to the global map.

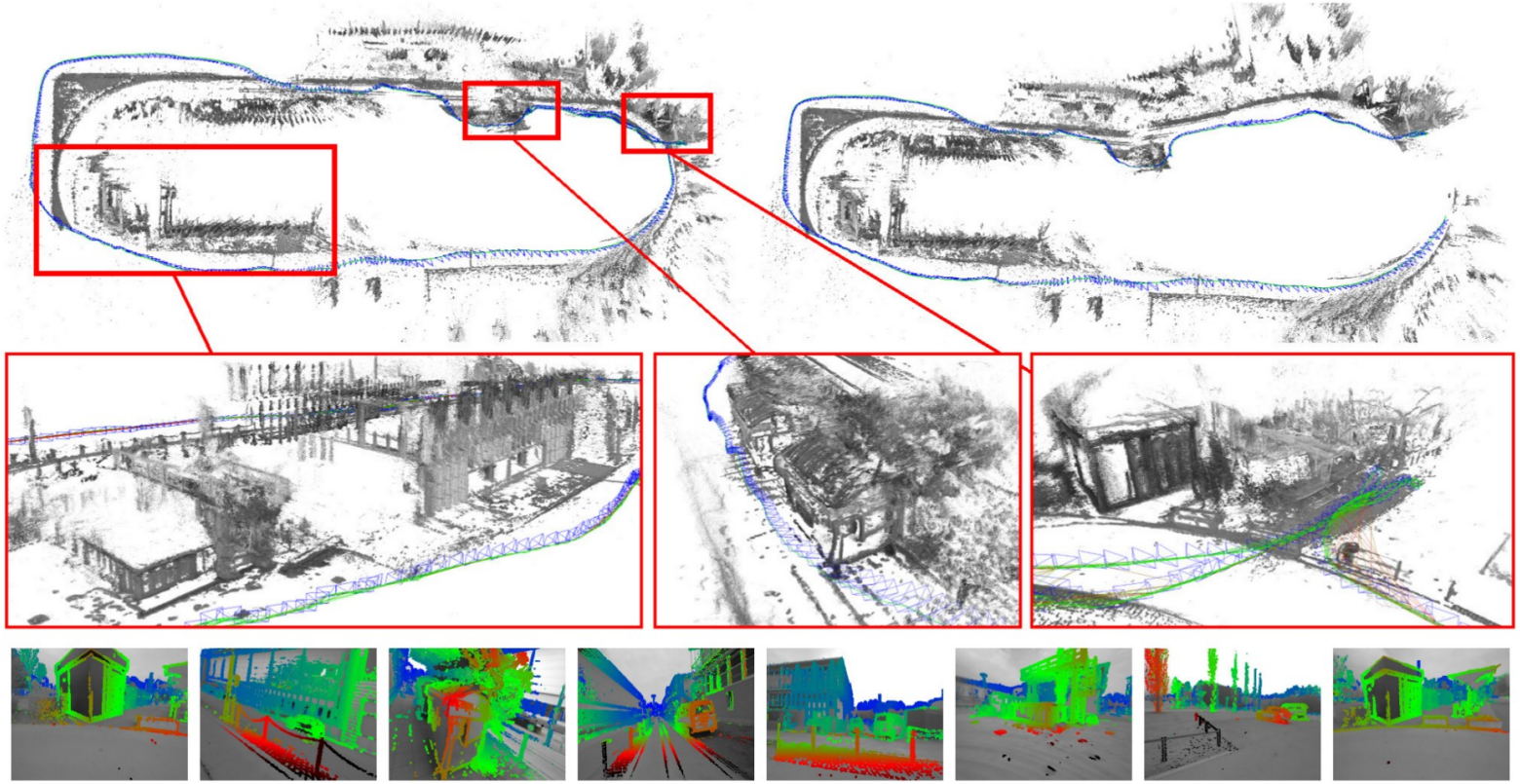


Engel, J., Sturm, J., & Cremers, D. (2013). Semi-dense Visual Odometry for a Monocular Camera. In 2013 IEEE International Conference on Computer Vision (pp. 1449– 1456)

Mapping

The map, consisting of a set of keyframes and tracked sim(3)-constraints, is continuously optimized in the background using pose graph optimization. The error function that is minimized is defined by

$$E(\boldsymbol{\xi}_{W1} \dots \boldsymbol{\xi}_{Wn}) := \sum_{(\boldsymbol{\xi}_{ji}, \boldsymbol{\Sigma}_{ji}) \in \mathcal{E}} (\boldsymbol{\xi}_{ji} \circ \boldsymbol{\xi}_{Wi}^{-1} \circ \boldsymbol{\xi}_{Wj})^T \boldsymbol{\Sigma}_{ji}^{-1} (\boldsymbol{\xi}_{ji} \circ \boldsymbol{\xi}_{Wi}^{-1} \circ \boldsymbol{\xi}_{Wj})$$



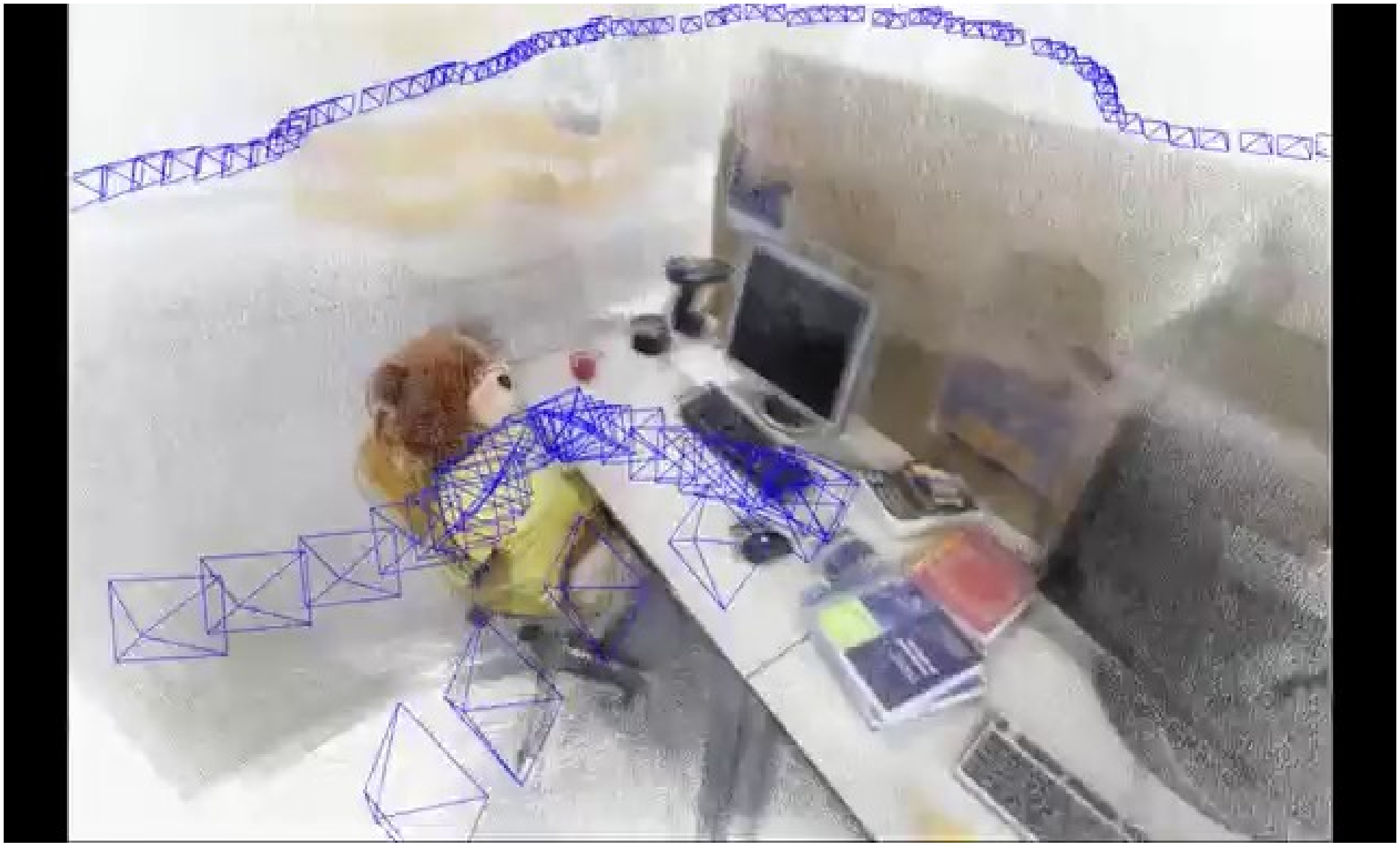
Engel, J., Sturm, J., & Cremers, D. (2013). Semi-dense Visual Odometry for a Monocular Camera. In 2013 IEEE International Conference on Computer Vision (pp. 1449– 1456)

Fig. 7: Loop closure for a long and challenging outdoor trajectory (after the loop closure on the left, before on the right). Also shown are three selected close-ups of the generated pointcloud, and semi-dense depth maps for selected keyframes.

ORB-SLAM

R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos,
“ORB-SLAM: A Versatile and Accurate Monocular SLAM System,”
IEEE Trans. Robot., vol. 31, no. 5, pp. 1147–1163, Oct. 2015.

ORB-SLAM



ORB-SLAM2 for Monocular, Stereo and RGB-D Cameras
Code: https://github.com/raulmur/ORB_SLAM2 .
Paper: Raúl Mur-Artal, and Juan D. Tardós. ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras, 2016

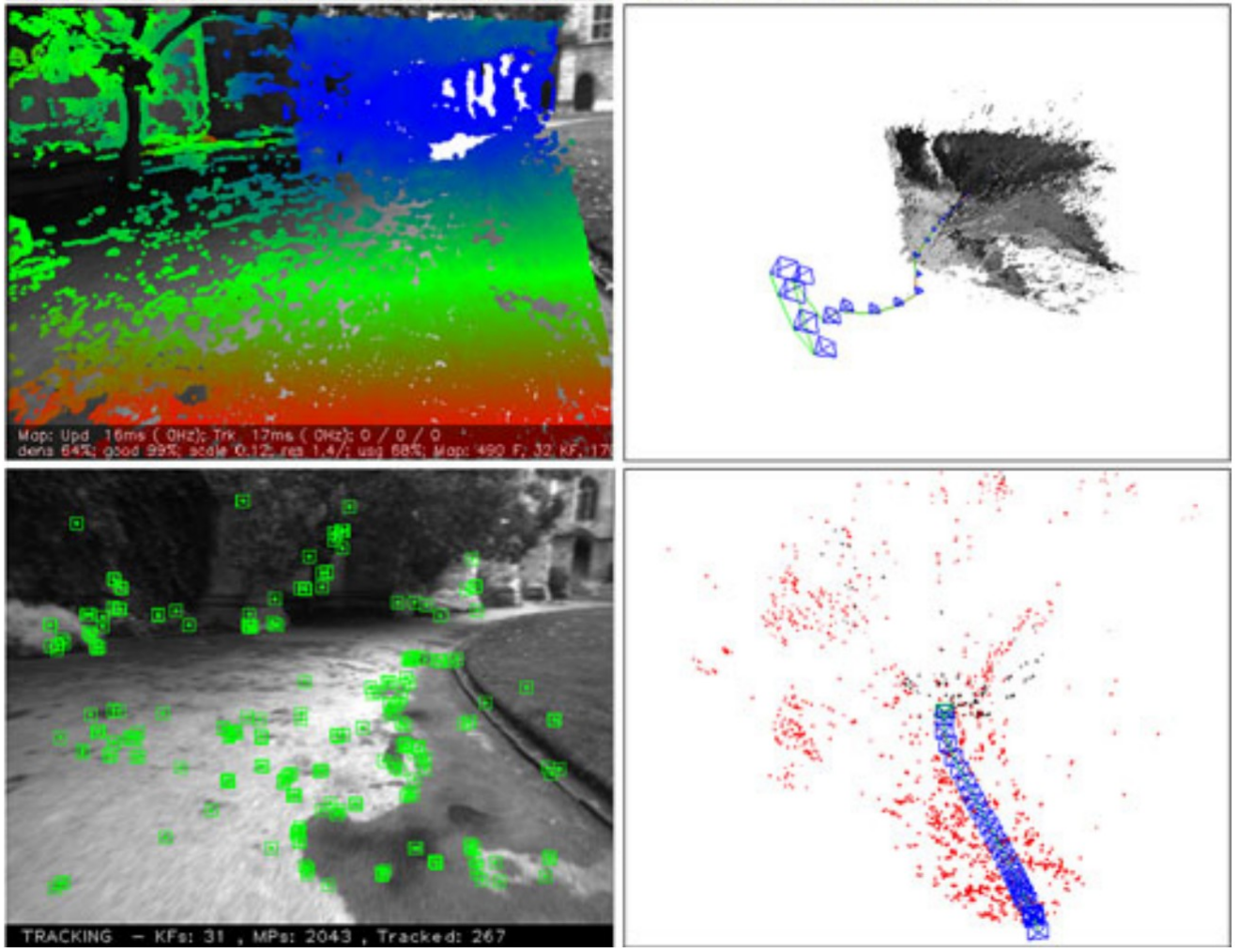
ORB-SLAM

Real-time SLAM must provide BA with

- Corresponding observations of scene features (map points) among a subset of selected frames (keyframes)
- As complexity grows with the number of keyframes, their selection should avoid unnecessary redundancy
- A strong network configuration of keyframes and points to produce accurate results, that is, a well spread set of keyframes observing points with significant parallax and with plenty of loop closure matches
- An initial estimation of the keyframe poses and point locations for the nonlinear optimization
- A local map in exploration where optimization is focused to achieve scalability
- The ability to perform fast global optimizations (e.g., pose graph) to close loops in real time

ORB-SLAM

R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos,
“ORB-SLAM: A Versatile and Accurate Monocular SLAM System,”
IEEE Trans. Robot., vol. 31, no. 5, pp. 1147–1163, Oct. 2015.



LSD-SLAM

ORB-SLAM

ORB-SLAM

Main contributions

- Use of the same features for all tasks:

  tracking, mapping, relocalization, and loop closing

- Real-time operation in large environments.

  Thanks to the use of a co-visibility graph, tracking and mapping are focused in a local co-visible area,independent of global map size

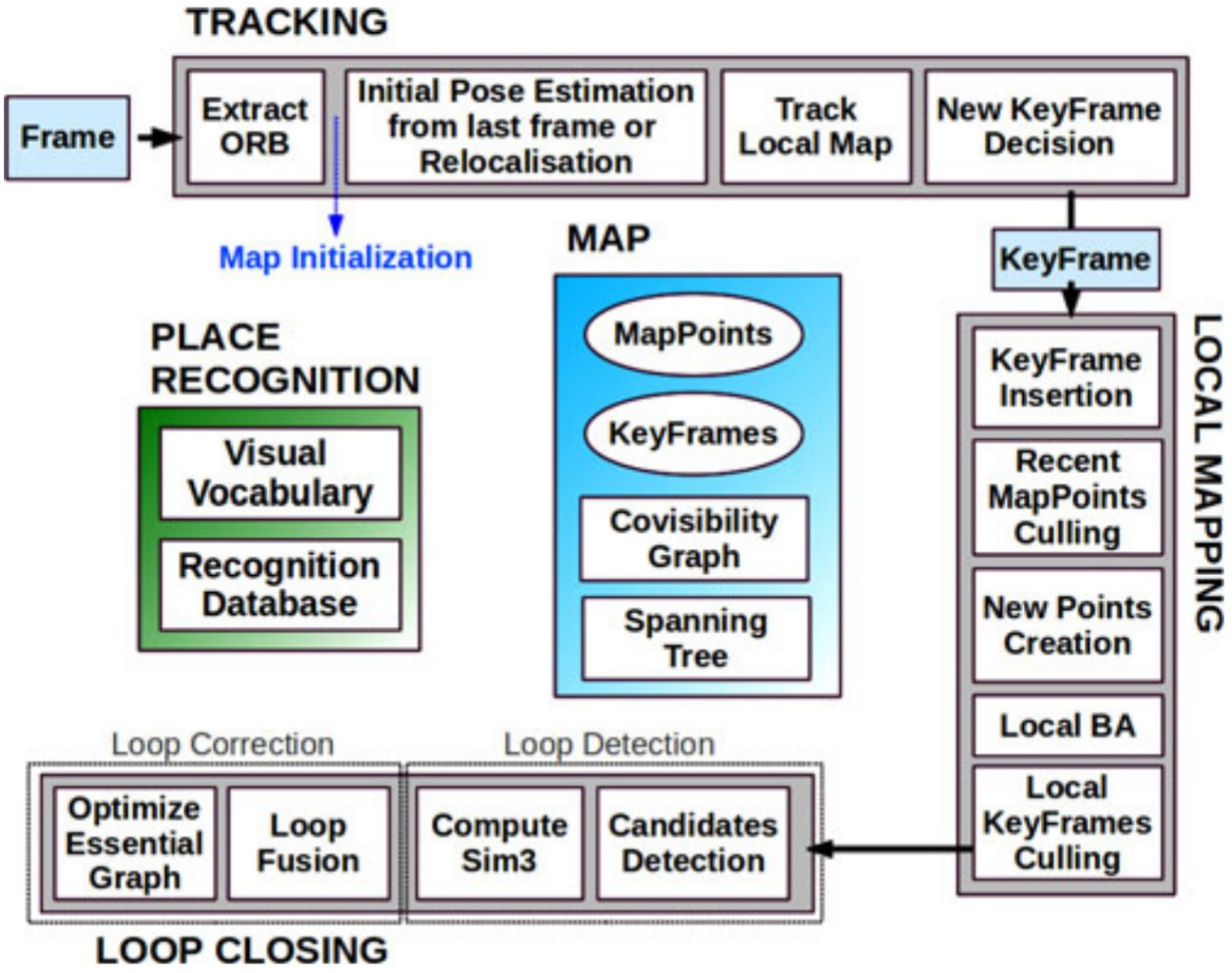
- Real-time loop closing based on the optimization of a pose graph that we call the Essential Graph

- Real-time camera relocalization with significant invariance to viewpoint and illumination. This allows recovery from tracking failure and also enhances map reuse

- A new automatic and robust initialization procedure based on model selection that permits to create an initial map of planar and nonplanar scenes

- A survival of the fittest approach to map point and keyframe selection that is generous in the spawning but very restrictive in the culling. This policy improves tracking robustness and enhances lifelong operation because redundant keyframes are discarded

ORB-SLAM System Overview



R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," IEEE Trans. Robot., vol. 31, no. 5, pp. 1147–1163, Oct. 2015.

Fig. 1. ORB-SLAM system overview, showing all the steps performed by the tracking, local mapping, and loop closing threads. The main components of the place recognition module and the map are also shown.

Short-term Tracking

- FAST corners in grid cells at different scale levels with ORB descriptors
- Initial pose estimation:
  - Tracking OK: Guided search with constant velocity model
  - Tracking ~OK: Wider search around positions in previous frame
  - Tracking lost: Global relocalization (long-term tracking)
- Track local map
  - Project local map and search for more correspondences
  - Motion-only bundle adjustment

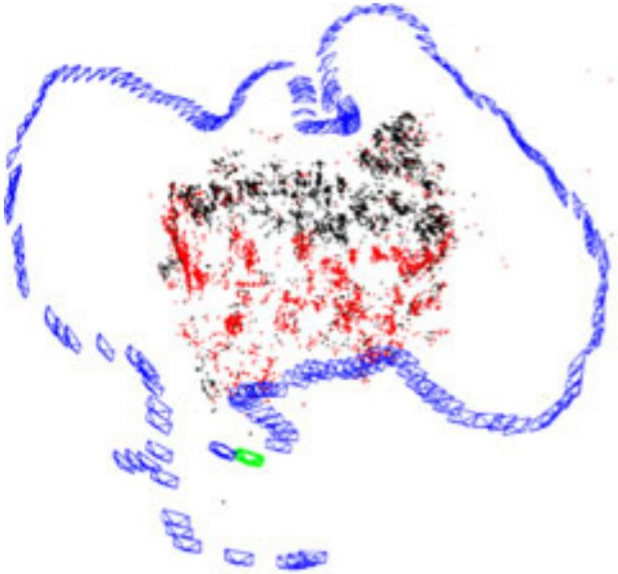
Keyframes

Insert keyframes often to make tracking more robust to rotations

- More than 20 frames since last global relocalization
- Local mapping is idle, or more than 20 frames since last keyframe insertion
- Current frame tracks at least 50 points
- Current frame tracks less than 90% than that of the reference

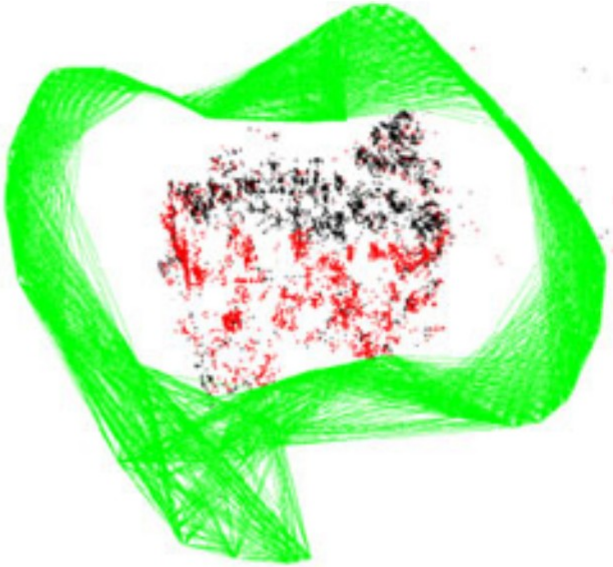
Map

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



Map co-visibility graph

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



R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," IEEE Trans. Robot., vol. 31, no. 5, pp. 1147–1163, Oct. 2015.

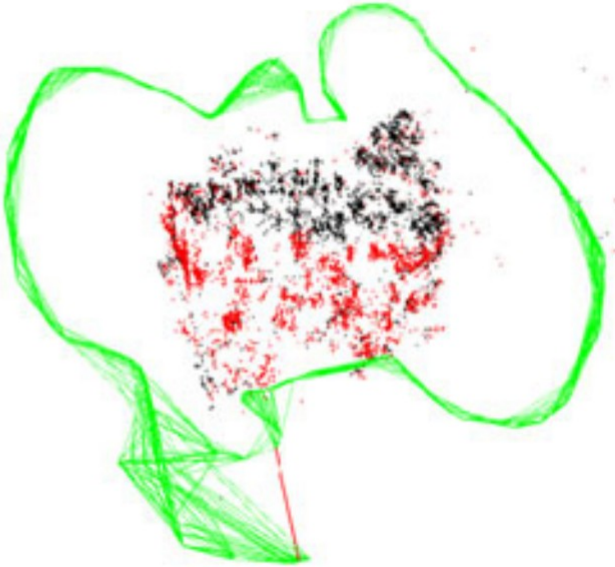
Map

Spanning tree

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

Essential graph

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



R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," IEEE Trans. Robot., vol. 31, no. 5, pp. 1147–1163, Oct. 2015.

Mapping

- Keyframe insertion
  - Add to co-visibility graph and spanning tree
- Recent map point culling
  - Remove bad points during the first three keyframes after creation
- New map point creation
  - Triangulate ORB from connected keyframes
- Local bundle adjustment
    - Optimize the current keyframe, all connected keyframes and all points seen
    - All other keyframes remain fixed
- Local keyframe culling
    - Detect and delete redundant keyframes

ORB-SLAM Long-term Tracking

- Long-term tracking is performed on the last keyframe in its own thread
- Loop closure candidates:
  - Compute BoW threshold based on the lowest score for the neighboring keyframes
  - Query recognition database (DBoW2) for keyframes with score higher than the threshold
  - Keep those that are not directly connected, and where we have at least three connected candidates
- Scale-drift aware loop closure alignment
  - Compute an initial similarity transform between the current keyframe and the loop keyframe from 3D-to-3D correspondences
  - Search for more correspondences
  - Optimize again
  - Geometric validation: Accept loop if enough inliers

ORB-SLAM Loop Correction

- Loop fusion
  - Fuse map points
  - Insert new edges in the co-visibility graph
- Essential graph optimization
  - Distribute the loop closing error along a pose graph
  - Transform each map point according to the correction of one of the keyframes that observes it

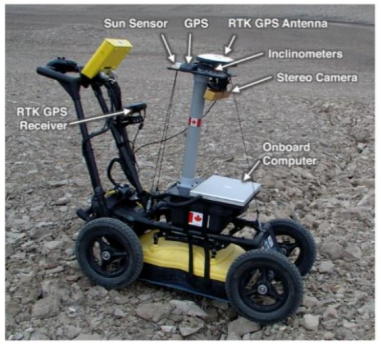
R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," IEEE Trans. Robot., vol. 31, no. 5, pp. 1147–1163, Oct. 2015.

State-of-the-art Open Source

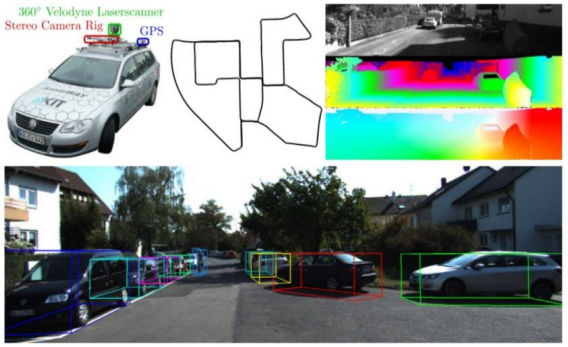
- MonoSLAM
- PTAM
- MSCKF
- DTAM
- SemiDense VO
- ORB-SLAM 2
- SVO
- LSD-SLAM
- OKVIS
- DSO

Popular Datasets for VO/SLAM Benchmarking

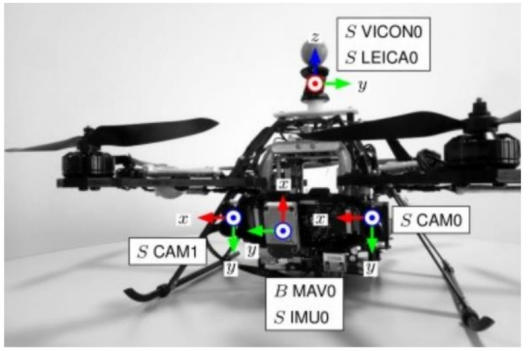
Devon Island [Furgale'11]
Stereo + D-GPS + inclinometer + sun sensor



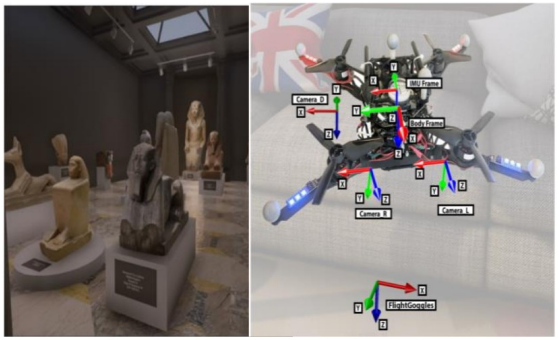
KITTI [Geiger'12]
Automobile, Laser + stereo + GPS, multiple tasks



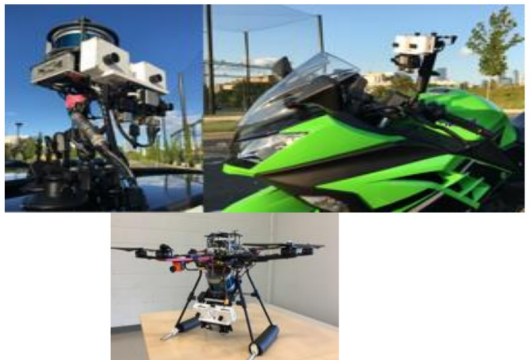
EuRoC [Burri'16]
MAV with synchronized IMU and stereo



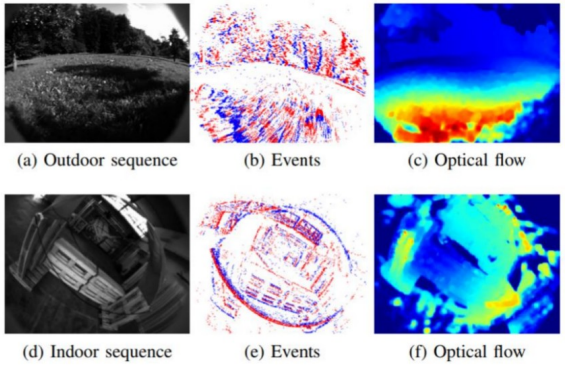
Blackbird [Antonini'18]
MAV indoor aggressive flight with rendered images and real dynamics + IMU



MVSEC [Zhu'18]
Events, frames, lidar, GPS, IMU from cars, drones, and motorcycles



UZH Drone Racing [Delmerico'19]
MAV aggressive flight, standard + event cameras, IMU, indoors and outdoors



D. Scaramuzza (UZH)

Want to know more?

Fordypmingsemne **TTK21 Introduction to Visual SLAM**

Lecturer: Trym Haavardsholm

Literature

- “Parallel Tracking and Mapping for Small AR Workspaces”,
  Klein and Murray,
  In Proc. International Symposium on Mixed and Augmented Reality (ISMAR'07, Nara), 2007
  https://www.robots.ox.ac.uk/~vgg/rg/papers/klein_murray__2007__ptam.pdf
- “Past, Present, and Future of Simultaneous Localization And Mapping: Towards the Robust-Perception Age”,
  Cadena et al., IEEE Transactions on Robotics 32 (6) pp 1309-1332, 2016
  https://ieeexplore.ieee.org/document/7747236
- “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