3D Computer Vision

Visual odometry primer and introduction

Ivan Marković

University of Zagreb Faculty of Electrical Engineering and Computing
Department of Control and Computer Engineering
Laboratory for Autonomous Systems and Mobile Robotics (lamor.fer.hr)

Outline

- Definition
- Modern odometry examples
  - Autonomous vehicles and Automated driver assistance systems
  - Autonomous mobile robots
  - Augmented/Virtual reality
  - Autonomus drones
  - Space exploratoin
- Visual odometry primer
  - Motivation
  - Building blocks
  - VO vs. VO-SLAM vs. SfM

Outline

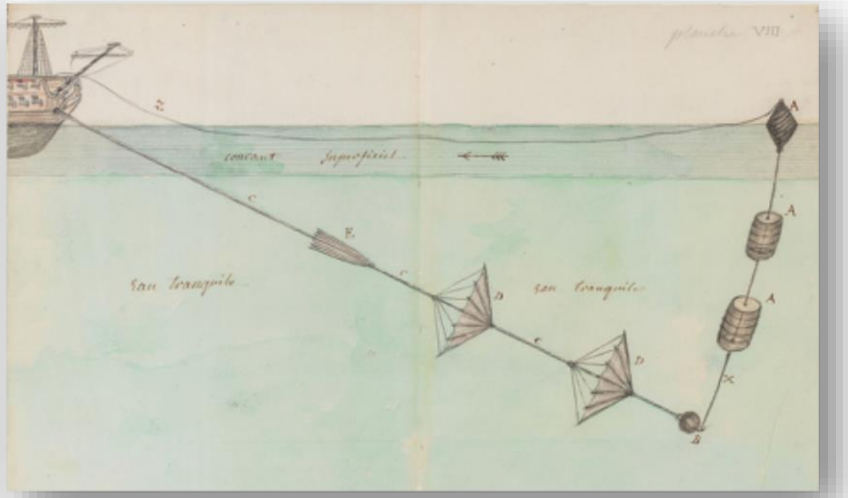
- Definition
- Modern odometry examples
  - Autonomous vehicles and Automated driver assistance systems
  - Autonomous mobile robots
  - Augmented/Virtual reality
  - Autonomus drones
  - Space exploratoin
- Visual odometry primer
  - Motivation
  - Building blocks
  - VO vs. VO-SLAM vs. SfM

What is odometry?

Ethimologically from ancient Greek:

hódos = „road, path, way”, métron = „to measure”

Another synonym is *dead reckoning*.

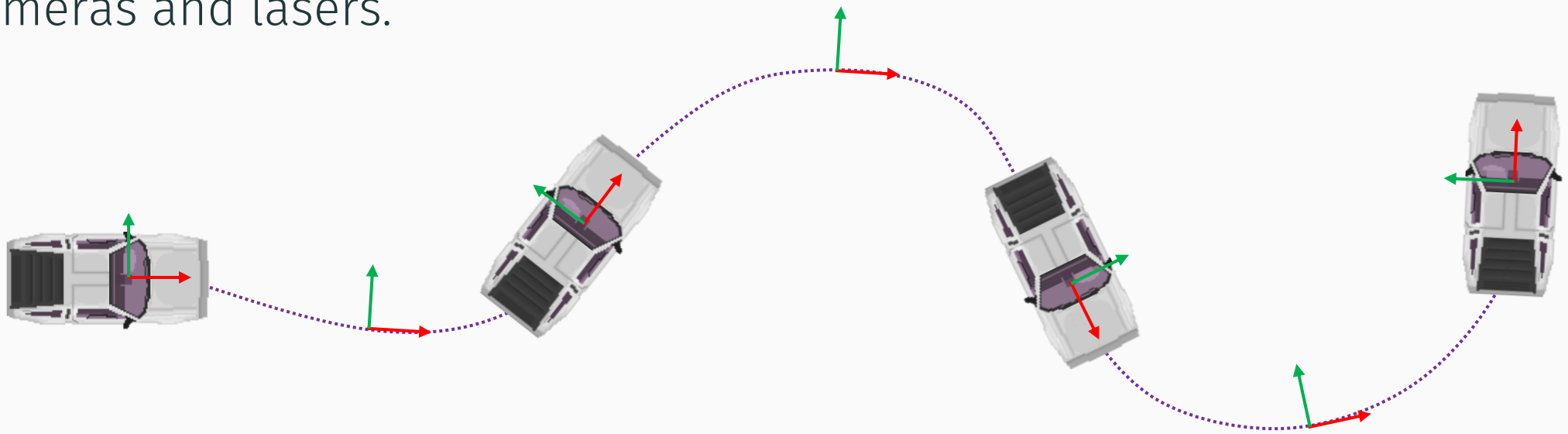


What is odometry?

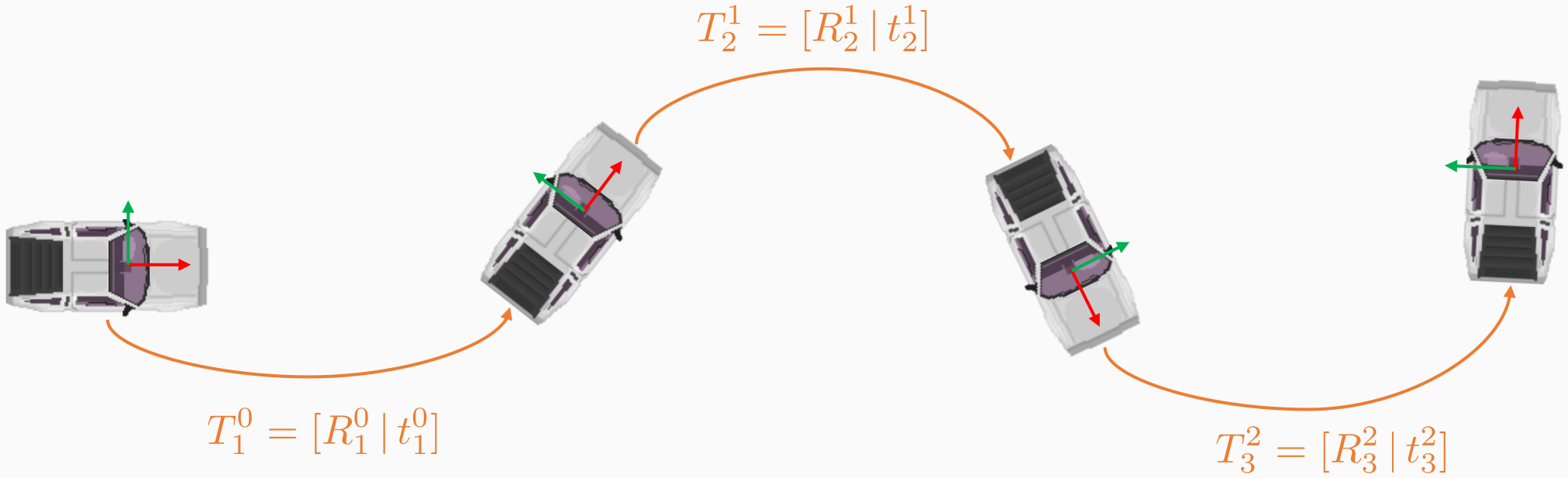
Specifically, we want to know where we are with respect to the starting pose, i.e., estimate the change in a pose (translation and orientation) over time.

Sensors moving freely in space or mounted on a moving platform, enable us to compute these small relative displacements that integrated over time yield the final pose in space. We often say that we estimate ego-motion.

Odometry can be estimated using proprioceptive sensors, e.g., wheel encoders, inertial measurement units, and exteroceptive sensors, e.g., cameras and lasers.



What is odometry?



By chaining transforms that represent relative displacements of vehicle's poses, we can estimate vehicle's ego-motion, i.e., compute the odometry.

The final pose of the vehicle w.r.t. the starting pose „0" is then simply

$$T_3^0 = T_1^0 T_2^1 T_3^2.$$

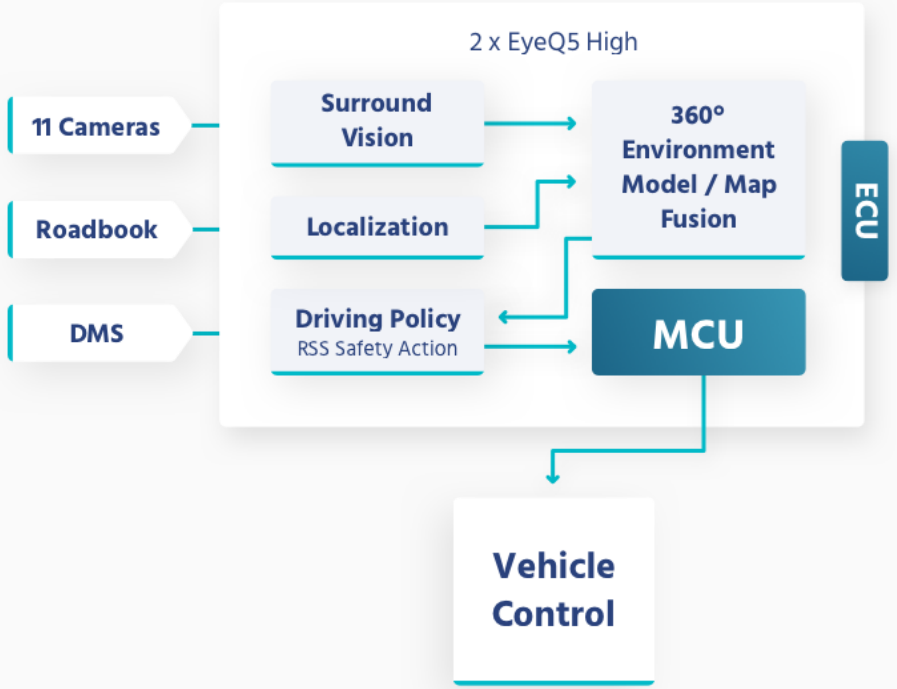
Outline

- Definition
- Modern odometry examples
  - Autonomous vehicles and Automated driver assistance systems
  - Autonomous mobile robots
  - Augmented/Virtual reality
  - Autonomus drones
  - Space exploratoin
- Visual odometry primer
  - Motivation
  - Building blocks
  - VO vs. VO-SLAM vs. SfM

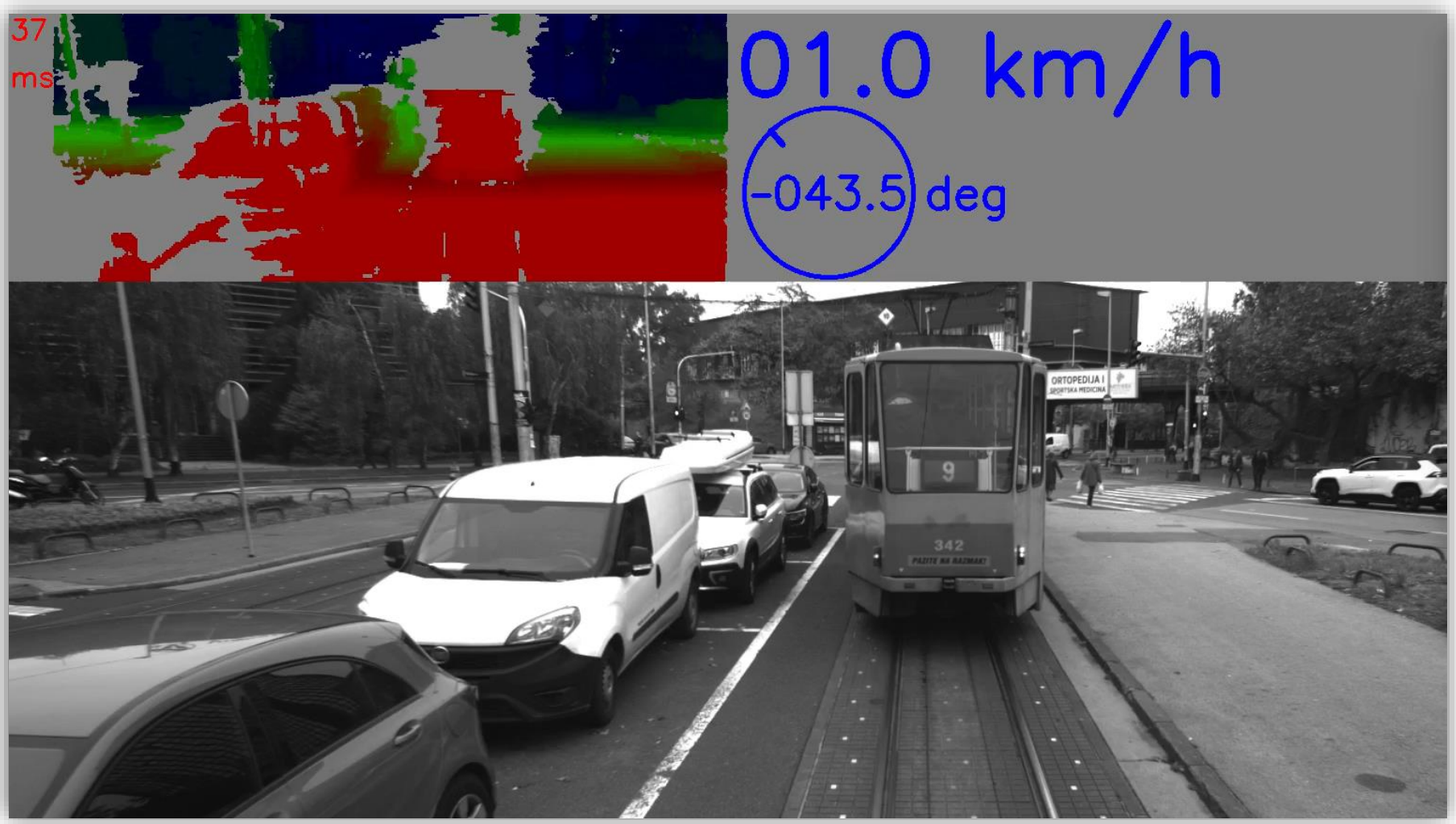
Autonomous driving and assistance systems

Mobileye SuperVision ADAS solution uses 11 cameras covering multiple features: hands-free driving (lane change, urban driving), self-parking, automatic preventive steering and braking.

All these systems require a plethora of computer vision algorithms as well as visual localization.

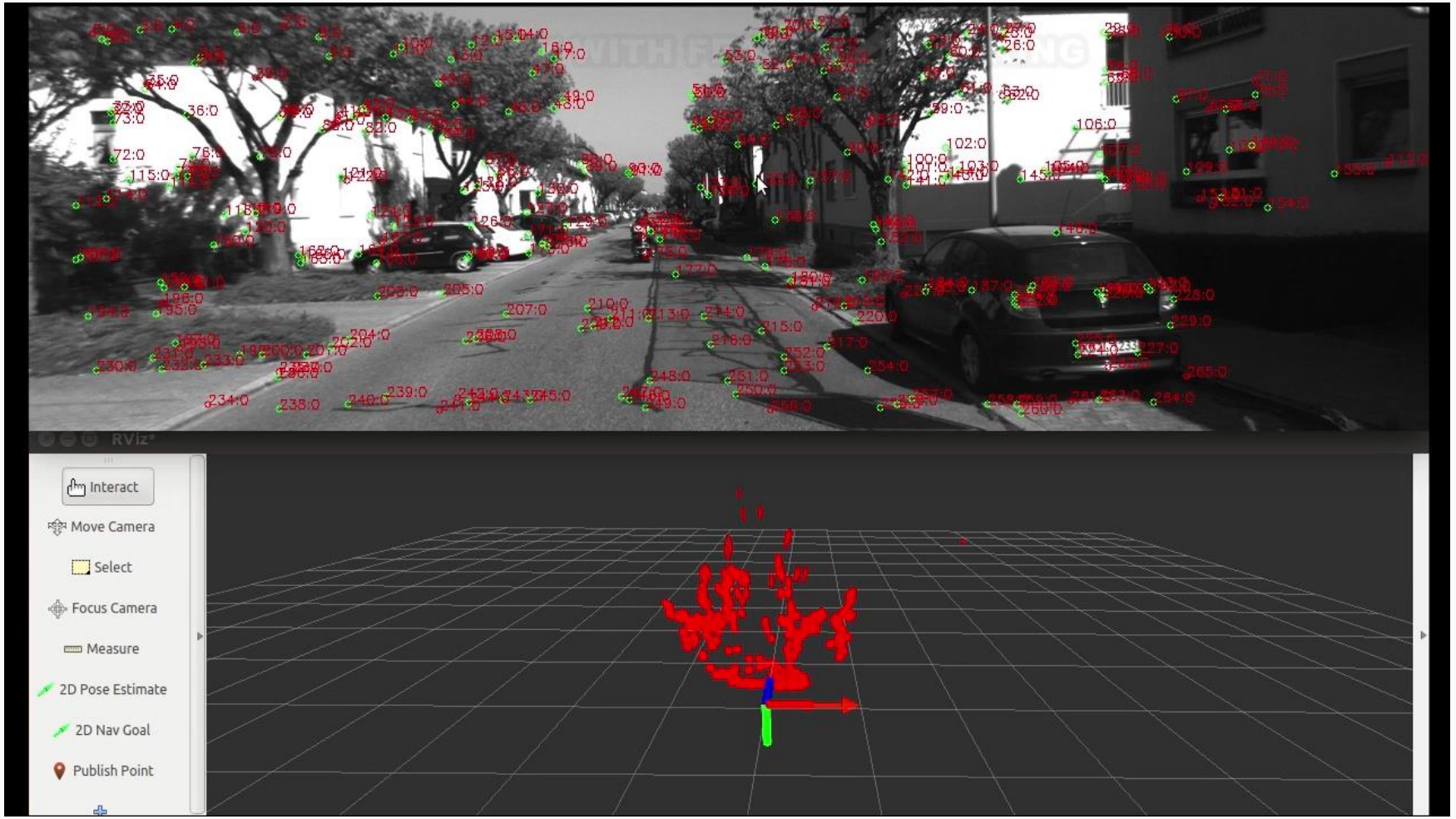


SafeTRAM project w/ Končar – forward collision warning



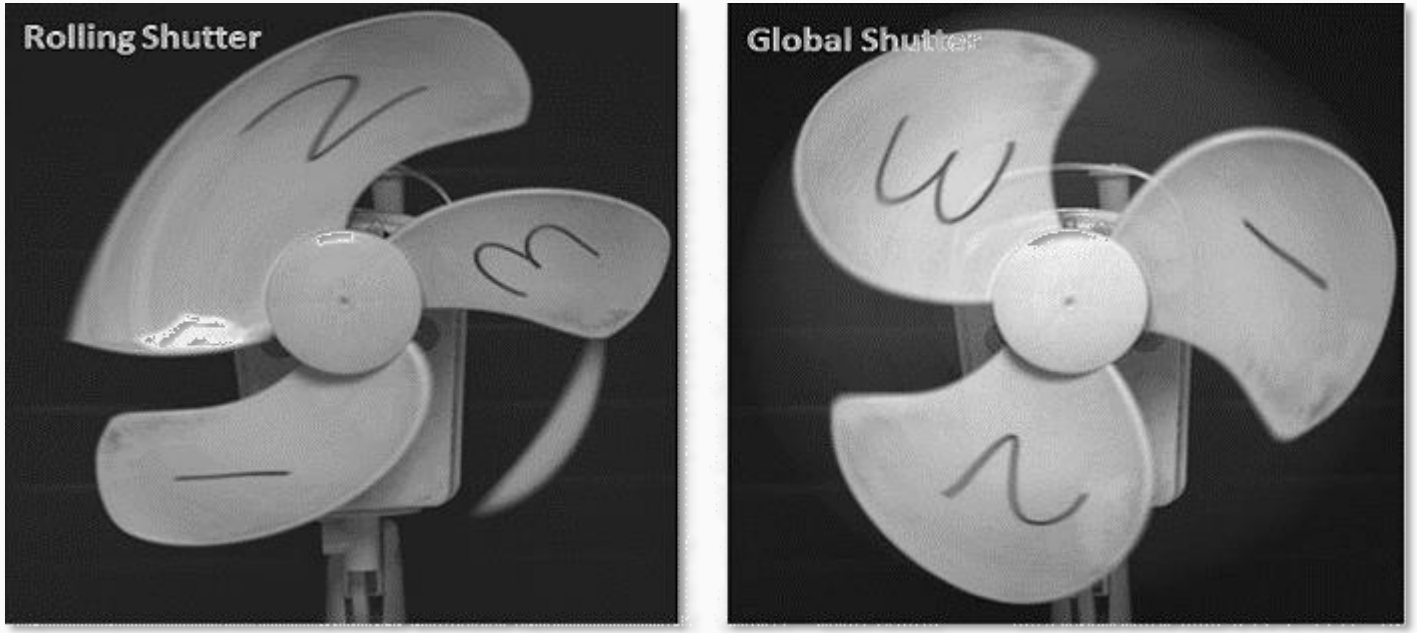
SOFT2 stereo odometry by Igor Cvišić

The highest ranking odometry on the KITTI benchmark.



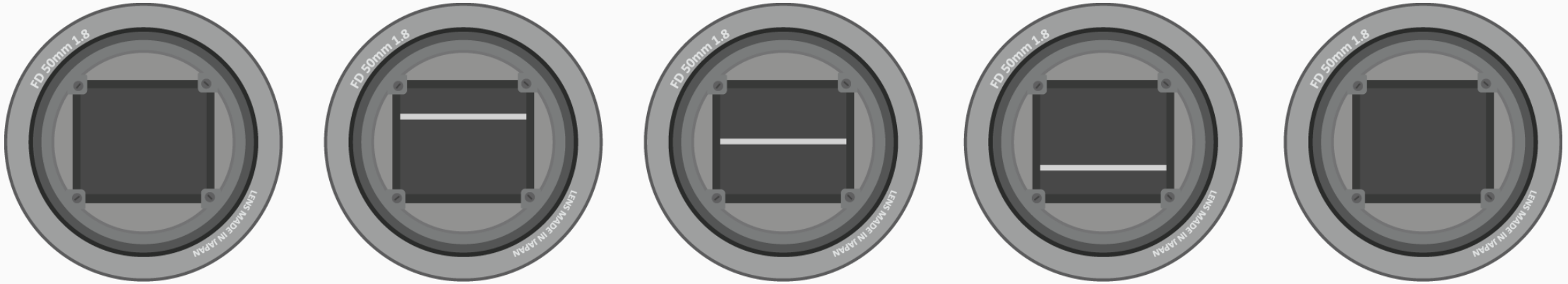
Rolling shutter vs. Global shutter

Rolling shutter captures the scene line-by-line which can cause distortion under dynamic camera/scene motion (autonomotive/mobile phone cameras are usually RS).



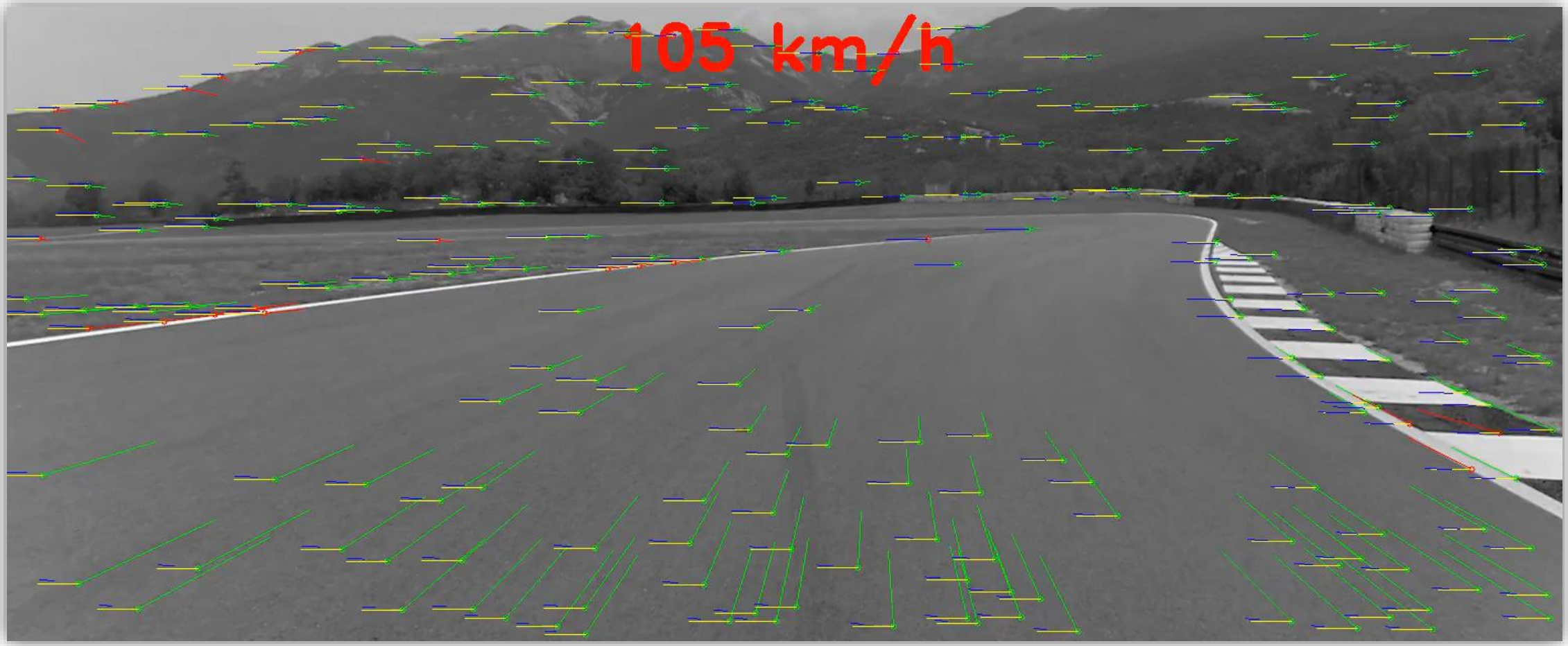
Global shutter captures the scene in a single take.

Rolling shutter illustration:



Source: BHPhotoVideo

SOFT2 adapted to rolling shutter for racing cars (w/ Rimac)

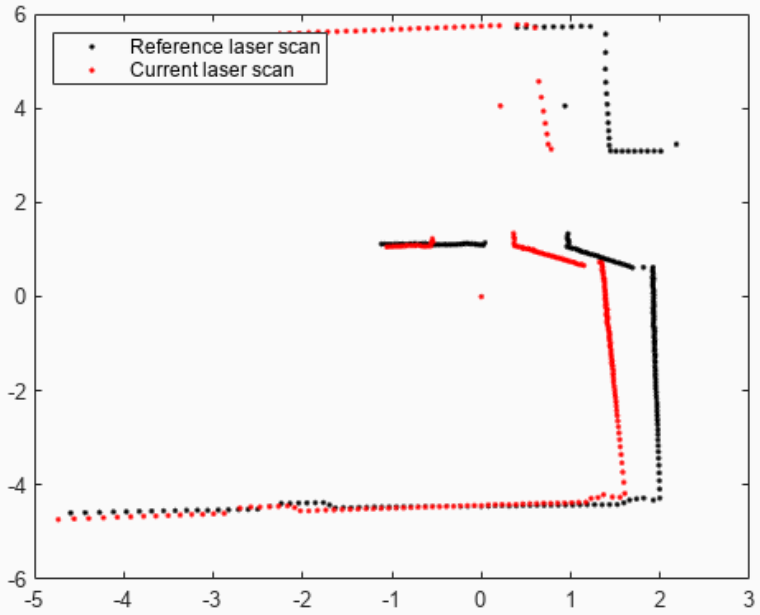


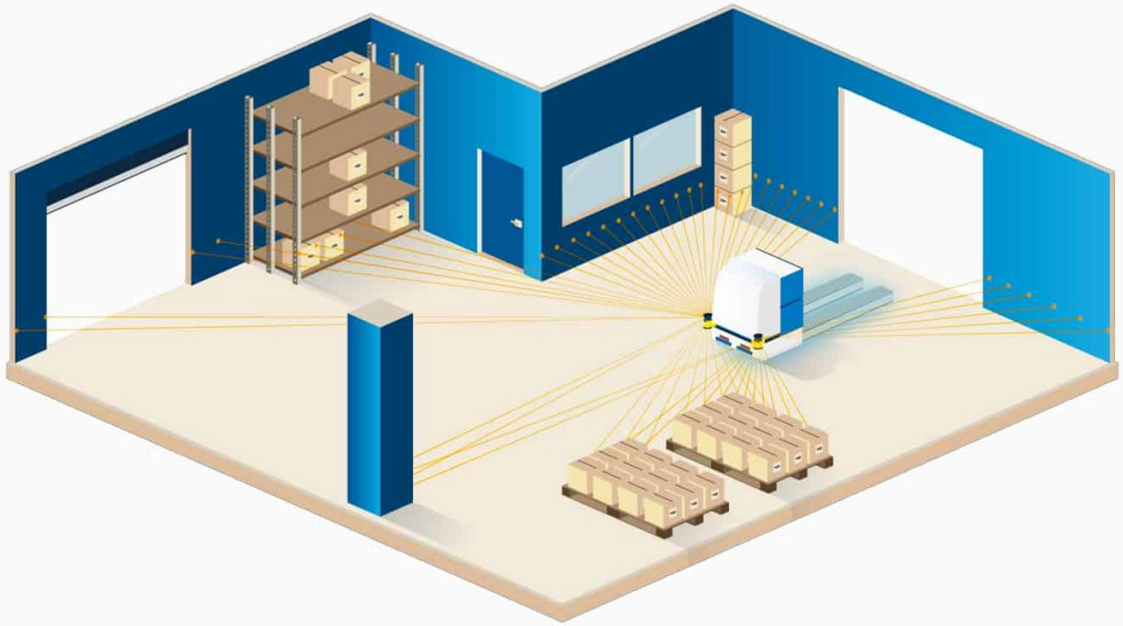
Mobile robot localization - scan matching

By consecutively aligning 2D laser scans we can estimate motion in a 2D plane. Used in mobile robotics (logistic robots and autonomous forklifts).

It can also be used for localization in a known map by cross-referencing current scans to a prebuilt map.

See this post from an industry pioneer, this seminal paper, and the paper with a point-to-line metric with exact close form solution.



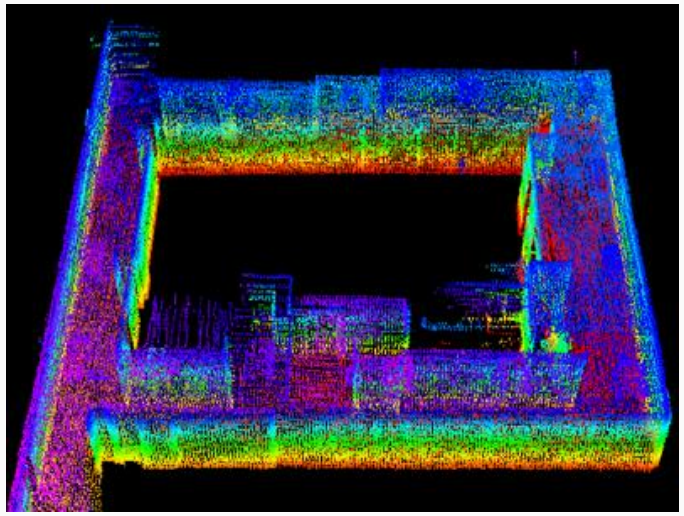


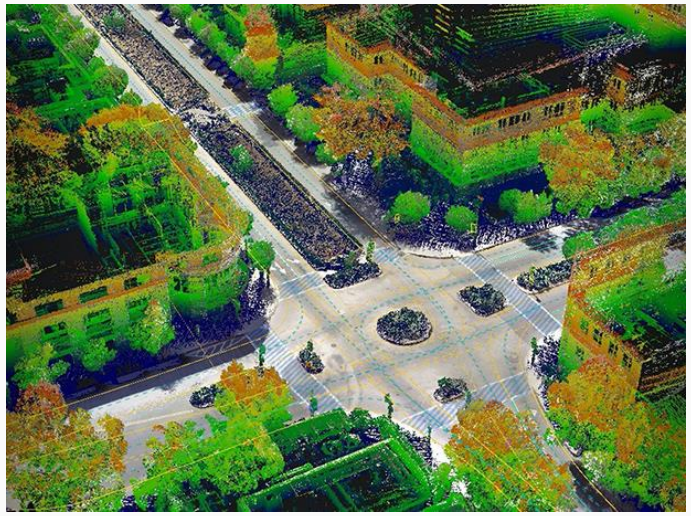
Odometry via 3D LiDAR ICP

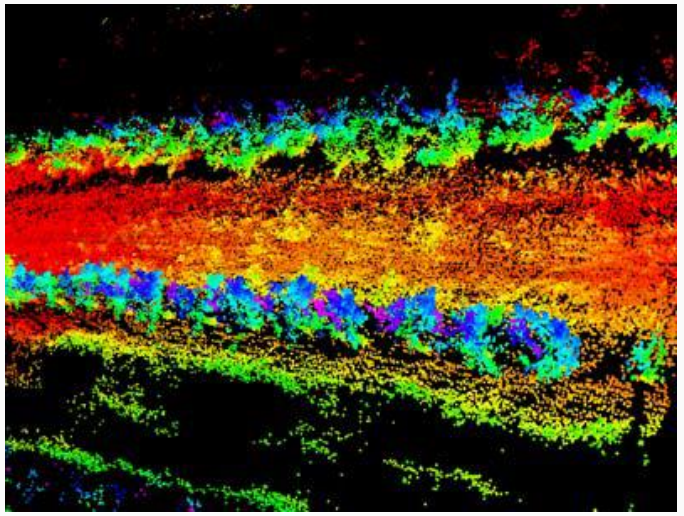
We can also use 3D registration techniques to estimate ego-motion from 3D LiDAR scans: point-to-point, point-to-plane, plane-to-plane error minimization (see the Generalized ICP algorithm).

See this seminal paper that was one of the early state-of-the-art baselines in robotics and autonomous vehicles.

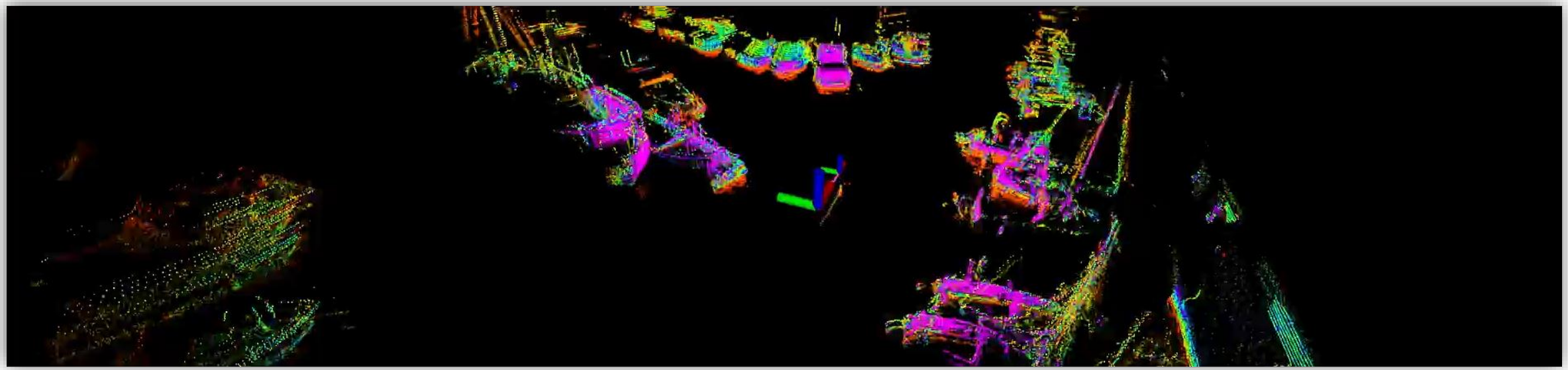
Enables creation of high-definition maps for autonomous vehicles (Nvidia DRIVE Map).







Marina 3D mapping (w/ LABUST)



cloudSLAM project - distributed mapping w/ 3D lasers

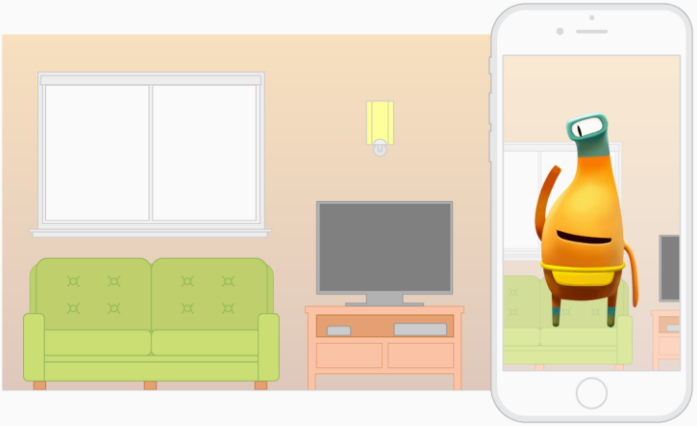
Augmented reality

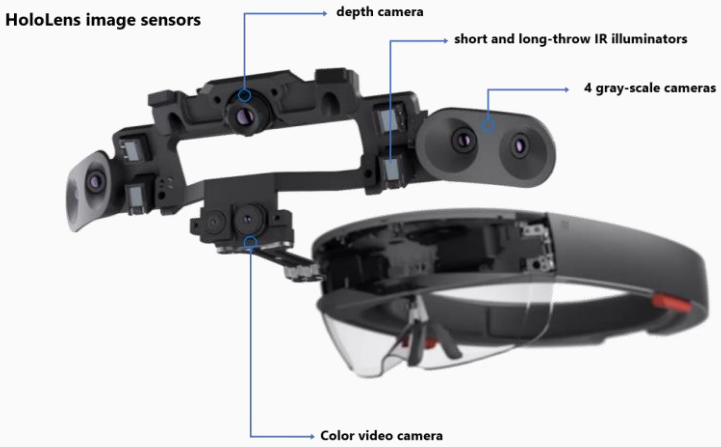
Visual odometry is a core component of many augmented reality implementations.

Google's ARCore relies on it, see the documentation and the patent.

Microsoft Hololens, that has 4 grayscale cameras, a depth camera and color camera, also runs it - see the documentation and the patent.

Apple's ARKit spatial tracking is reliant on it to understand the surrounding environment, see the documentation.

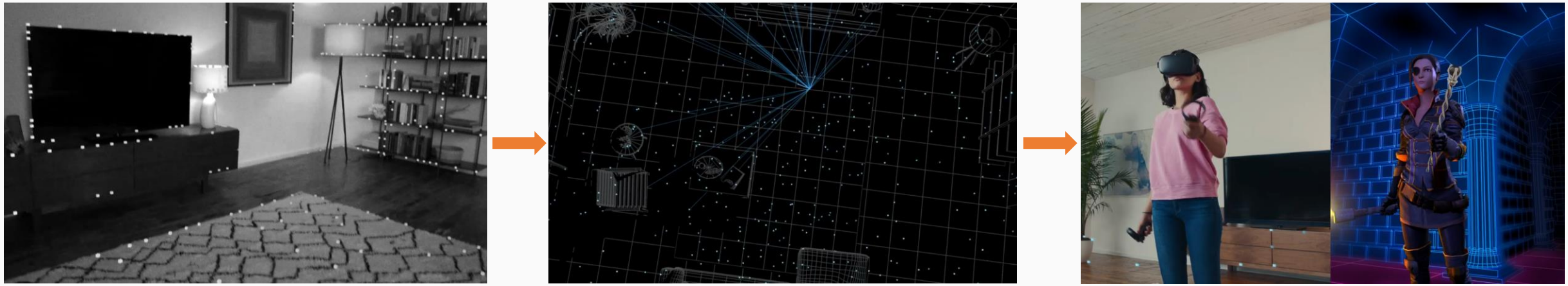




Virtual reality

It is also indispensable in virtual reality applications as accurate pose of the player's headset is a requirement for rendering immersive virtual environments anywhere.

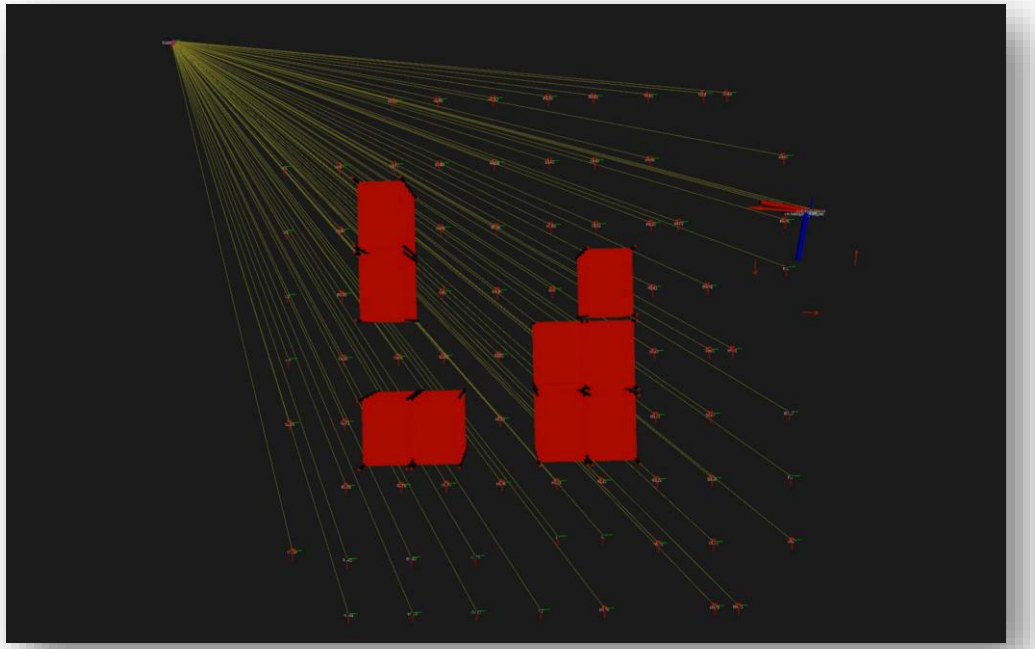
Read this Oculus' post on Oculus Insight: „ all-in-one, completely wire-free VR gaming system, we knew we needed positional tracking that was precise, accurate, and available in real time".



SafeLog project - human localization with wearable sensors

By mounting light sensors on a human, we can localize workers in various environments, e.g., robotized warehouses.

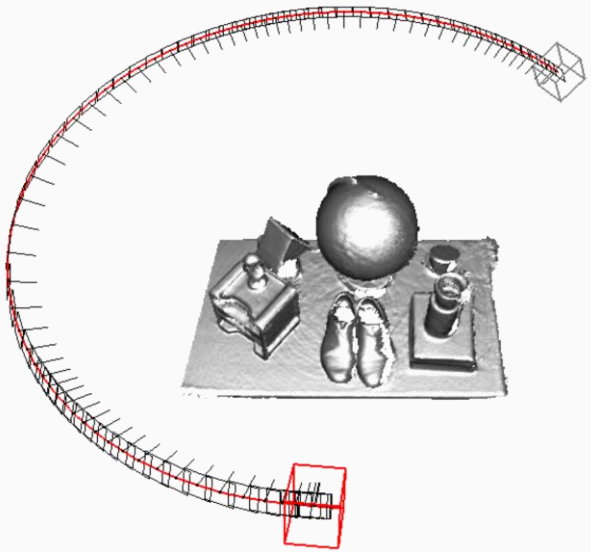
This can increase the safety of the whole system and efficiency of robot deliveries, since the fleet can be coordinated in a way to avoid intesecting robot paths as much as possible with the humans'.

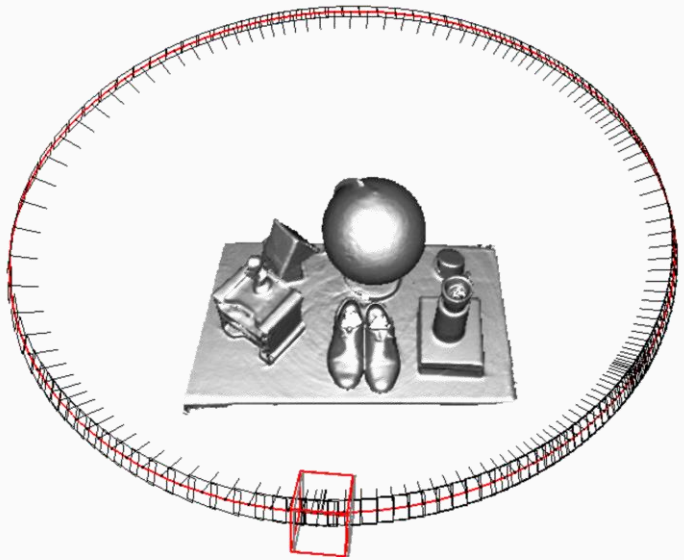


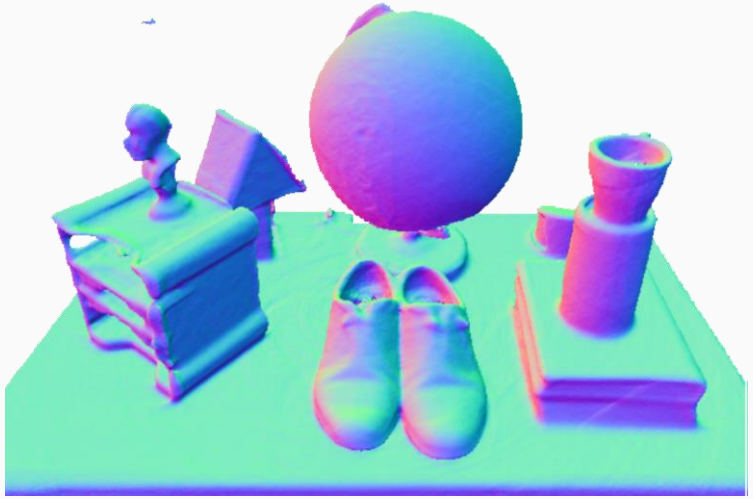
Odometry and mapping with RGBD sensors

Data from RGBD cameras can also be used to estimate ego-motion and map the environment.

Estimated ego-motion can also be used to construct dense object models. See one of the first real-time algorithms developed for Microsoft Kinect v1.



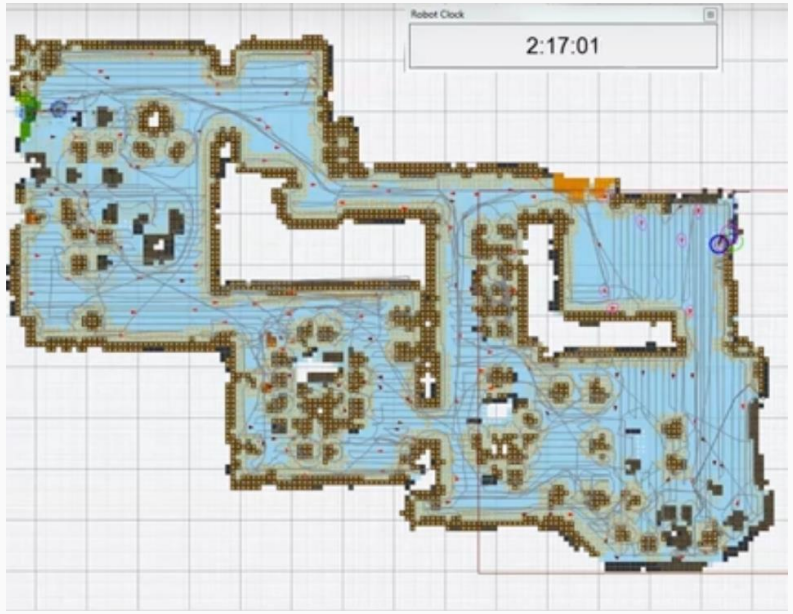




Household robots

Visual localization, which includes visual odometry, enables the new generation Roombas to follow predefined paths instead of navigating randomly across the rooms.

This ensures guaranteed coverage and lower vacuuming time.



Drone autonomy

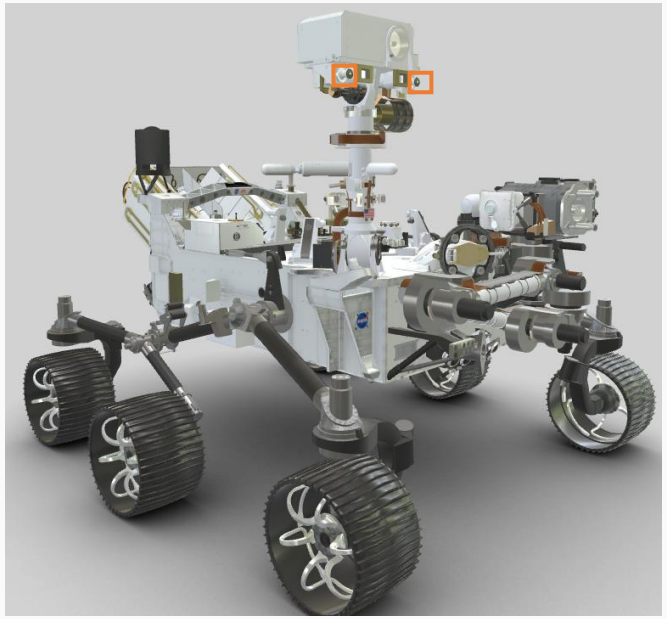
DJI drones use multiple stereo cameras and visual odometry to localize and fly, follow people and avoid obstacles.

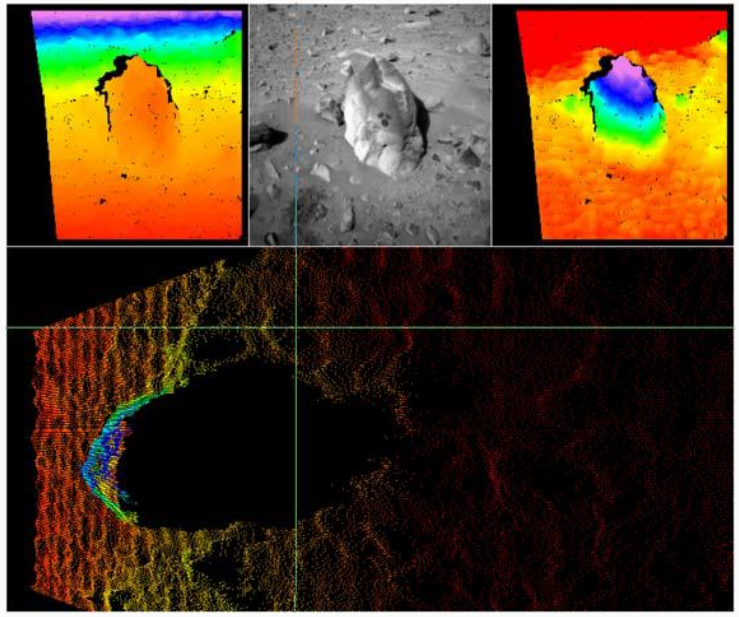
DJI Guidance system can be acquired separately and used for other applications, or within the developers drone Matrice series.

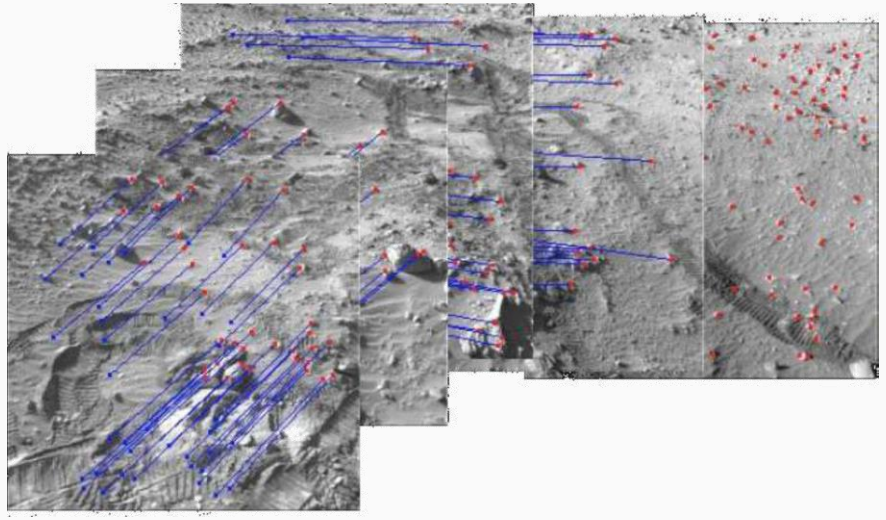
Mars exploration rovers

NASA’s rovers since the first missions relied on visual odometry for navigation and a stereo pair of navigation cameras is an indispensable part of rovers (Perseverance NavCams – left image).

See the following paper on visual odometry used in Spirit and Opportunity and this paper about computer vision on Mars (stereo depth of a Martian rock – middle, feature tracking on Mars – right image).





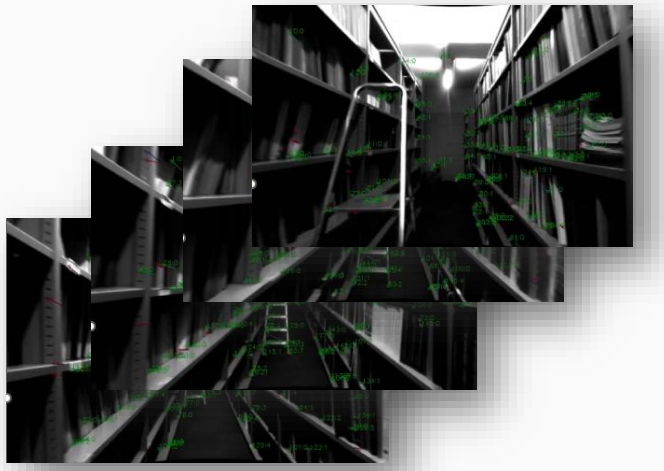


Outline

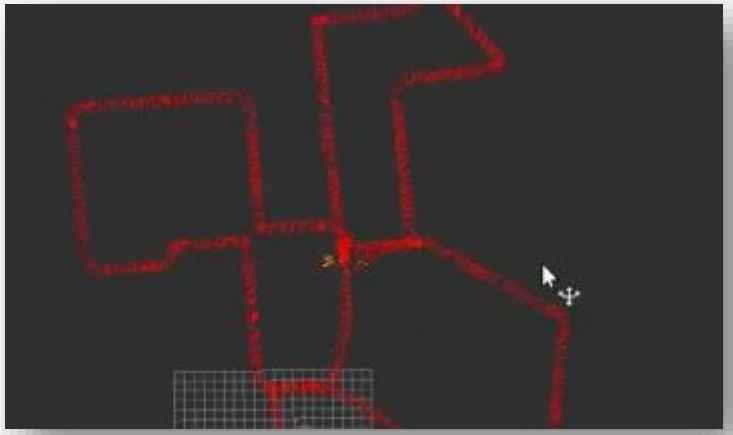
- Definition
- Modern odometry examples
  - Autonomous vehicles and Automated driver assistance systems
  - Autonomous mobile robots
  - Augmented/Virtual reality
  - Autonomus drones
  - Space exploratoin
- Visual odometry primer
  - Motivation
  - Building blocks
  - VO vs. VO-SLAM vs. SfM

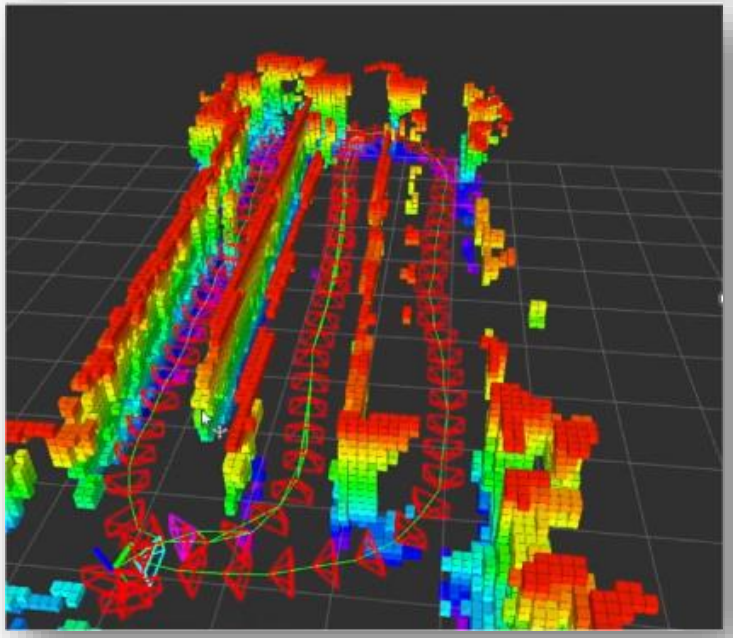
Visual odometry input and output

Input: a sequence of images



Output: sensor trajectory





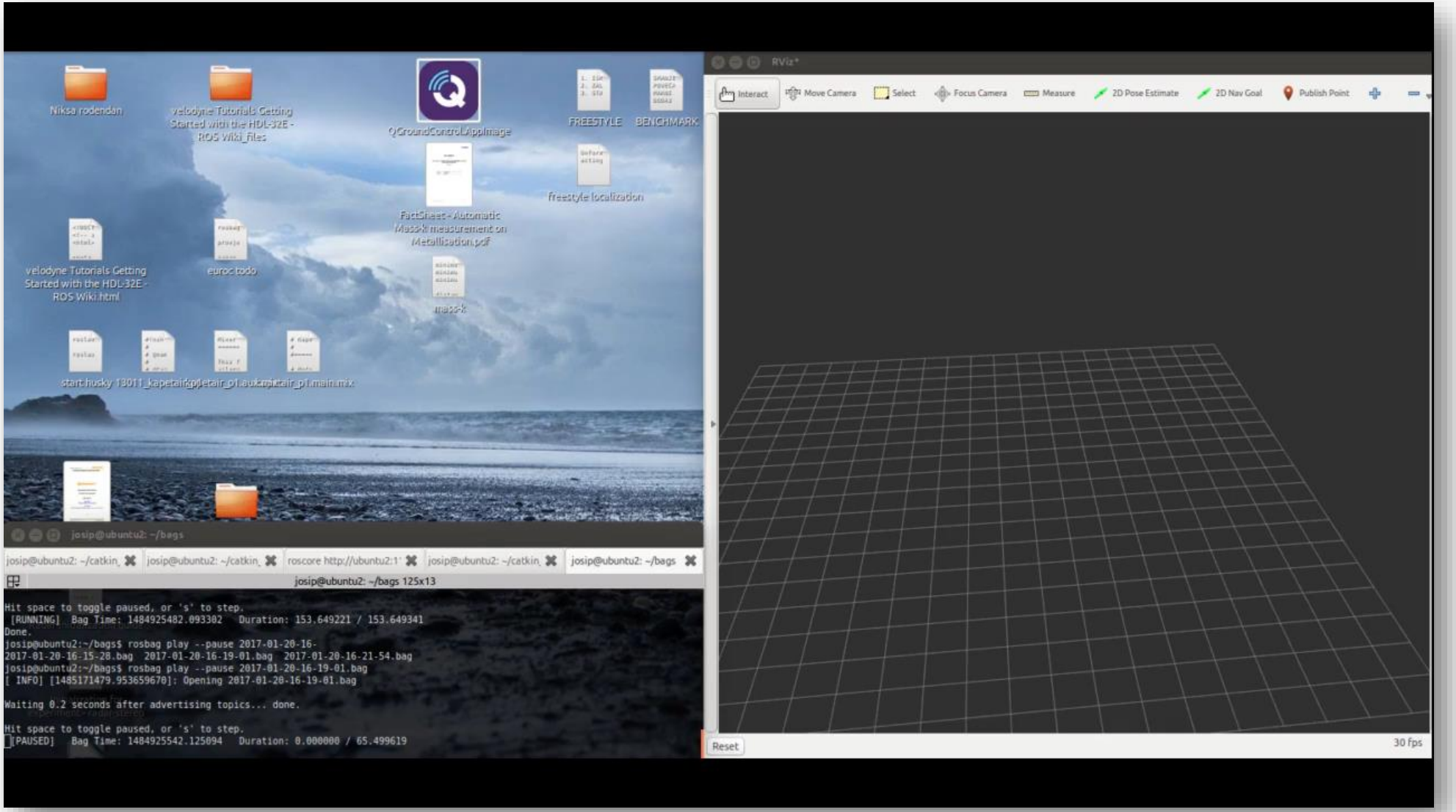
Sequence of relative pose transformations:

$$[R_1^0 \mid t_1^0], [R_2^1 \mid t_2^1],$$

$$\ldots, [R_j^i \mid t_j^i], \ldots$$

(environment can also be constructed either roughly or in details).

Visual odomety in FER's library



Why visual odometry?

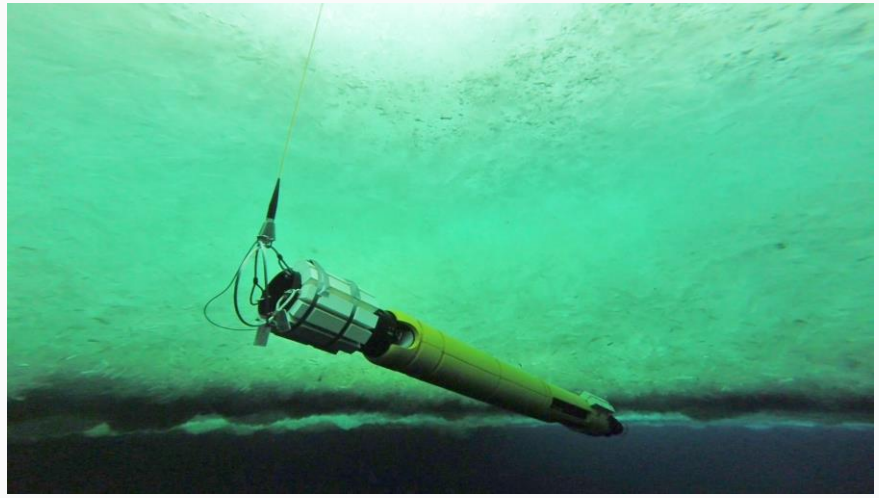
Cameras are relatively inexpensive sensors that yield rich information, thus besides odometry their input can also be used for other purposes, e.g., objects detection, semantic segmentation, depth perception etc.

Modern visual odometries achieve relative translation errors between 0.5% and 1% of the total traversed path (see the KITTI benchmark).

Autonomous systems that are expected to be robust couple visual odometry with other sensor modalities:

- Wheel odometry
- Inertial measurement units
- 2D and 3D LiDARs
- GNSS

Consider under-ice robot exploration.



A brief history

1913 Kruppa's proof of at most 11 solutions to the 5-point relative pose problem

1980 First real-time implementation of visual odometry by H. Moravec for NASA Mars rovers:

- single camera sliding on a rail horizontally, taking 9 equidistant images – still stereo vision
- corner detection with Moravec's algorithm and correlation

1981 Longuet-Higgins' 8-point algorithm for essential matrix estimation

1997 Hartley's normalized 8-point algorithm

- 2000 Research dominated by NASA/JPL for the mission to Mars (2004):

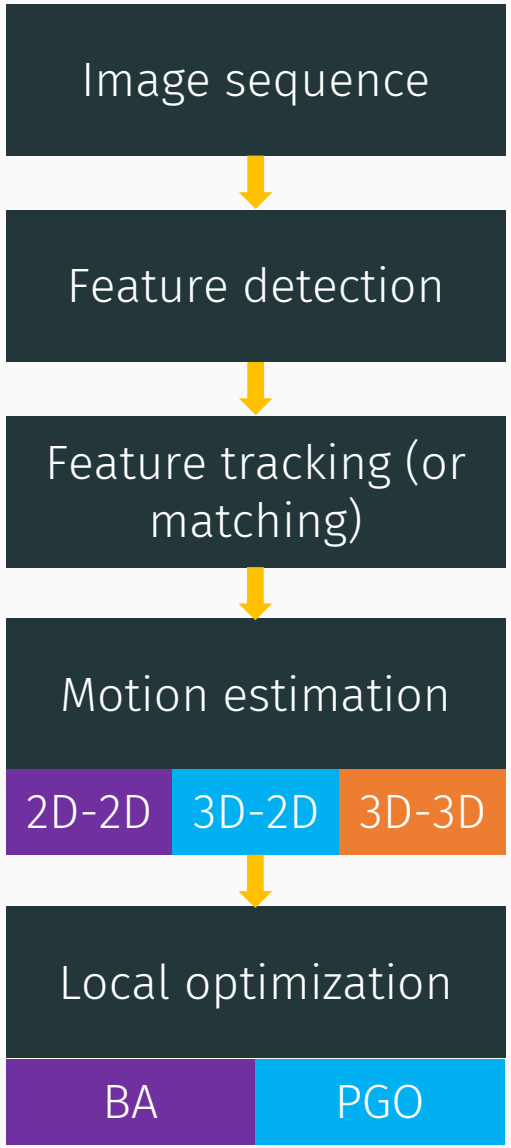
2004 Mission to Mars (Spirit and Opportunity):

- 2 grayscale cameras for stereo odometry and depth reconstruction

2004 Níster's 5-point algorithm for essential matrix estimation

- Visual odometry term introduced, revives interest in the academia

Building blocks of visual odometry



This can be a sequence of mono or stereo images

Corners, blobs, SIFT, SURF, ORB etc.

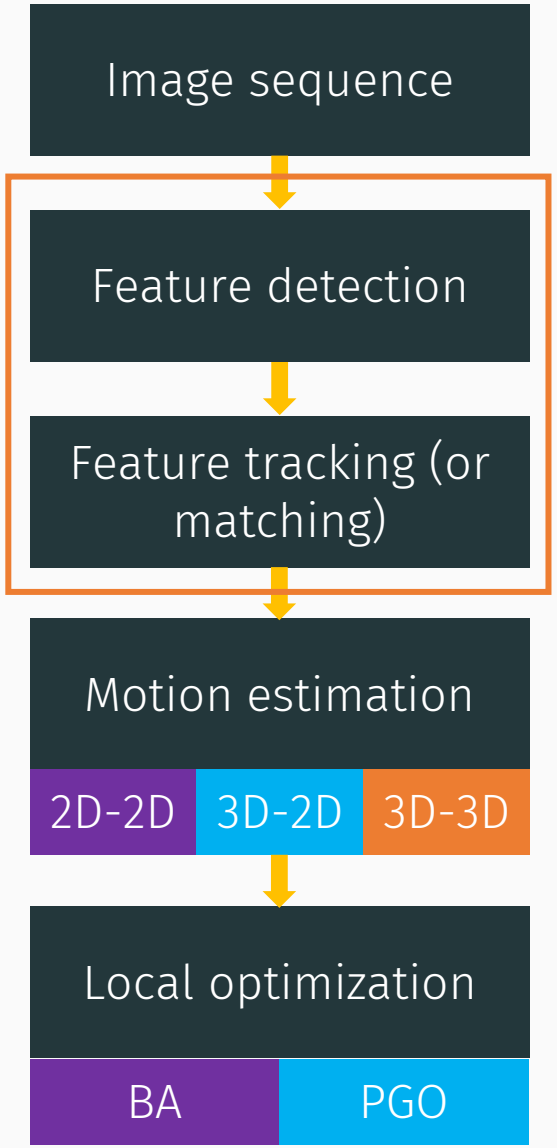
Optical flow, correlation, descriptor matching

Depending whether mono or stereo, various strategies are possible; however, note that mono can only estimate translation up to scale.

An optional step to carry out optimization over a window of several past frames to enforce local consistency:

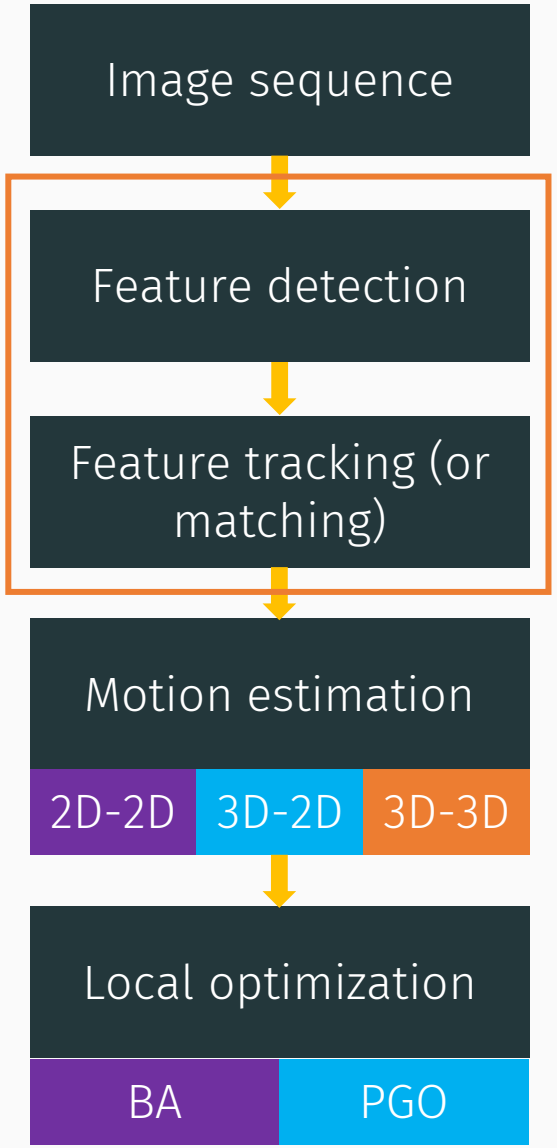
1. Bundle adjustement (BA) – optimizes over poses and features
2. Pose graph optimization (PGO) – optimizes only over poses

Building blocks of visual odometry



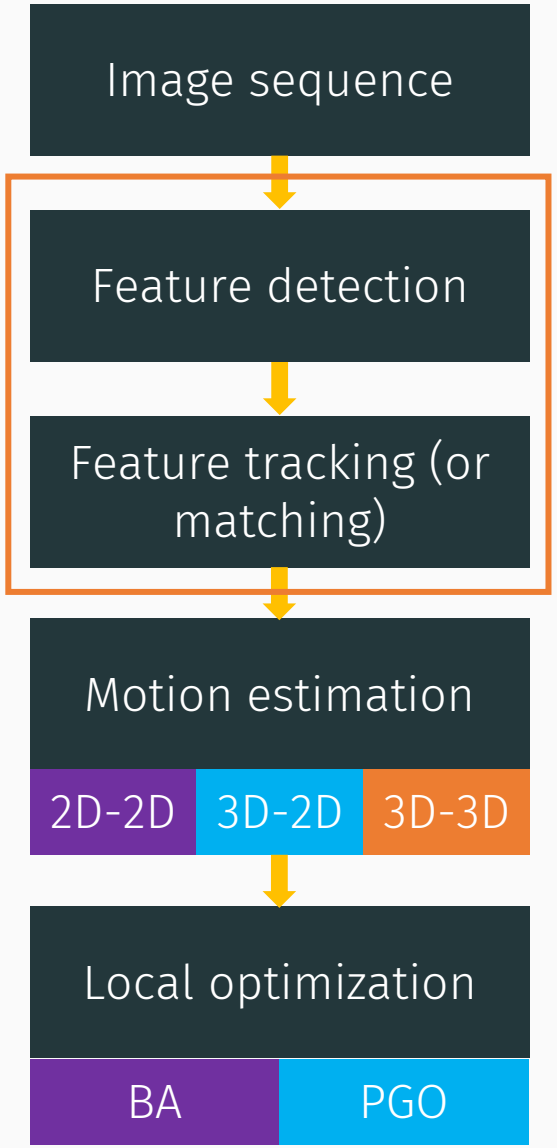
1. Detect features in the previous image

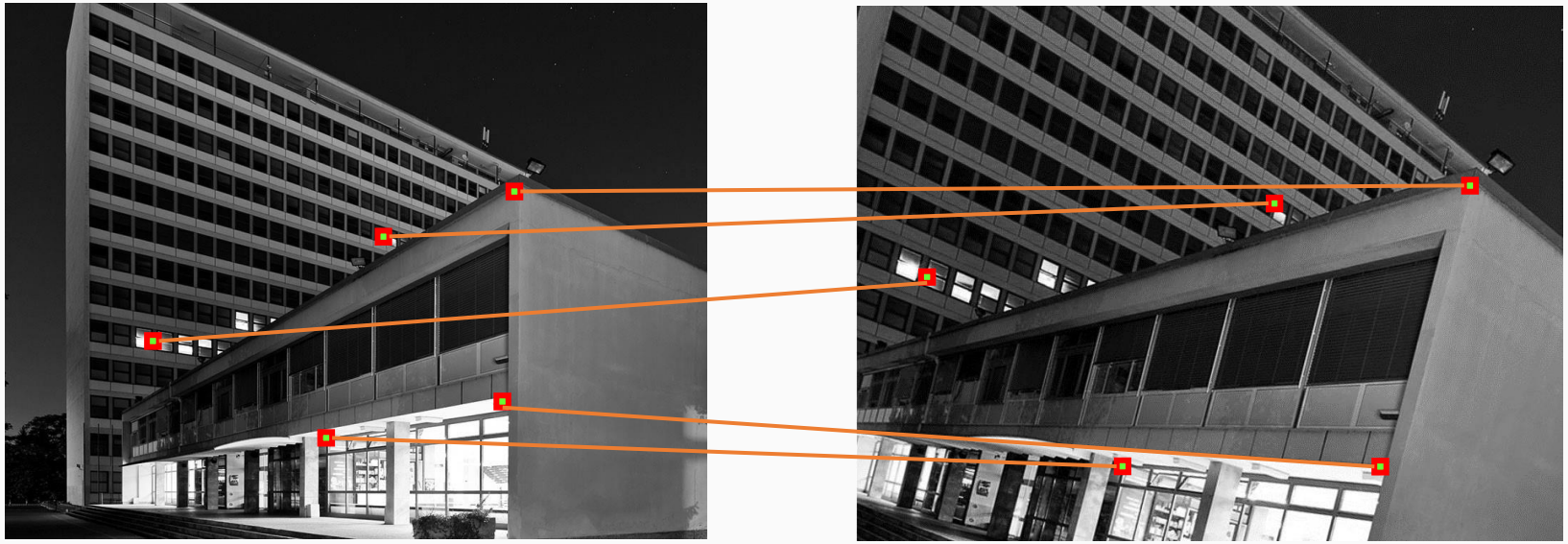
Building blocks of visual odometry



1. Detect features in the previous image
2. Detect features in the current image

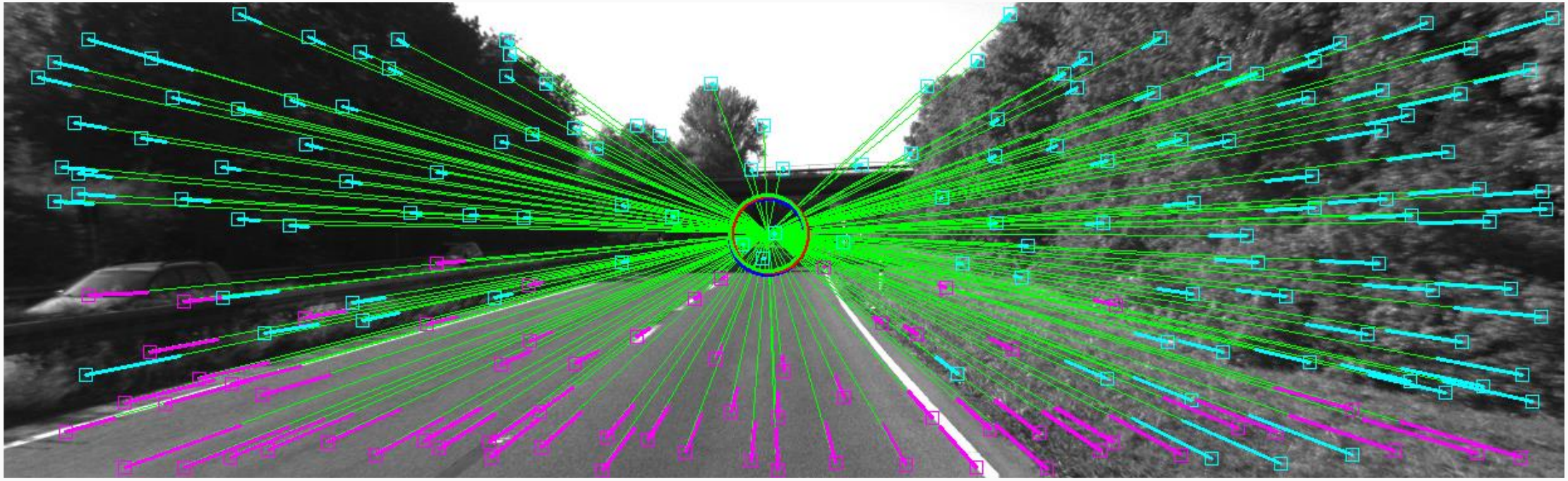
Building blocks of visual odometry





1. Detect features in the previous image
2. Detect features in the current image
3. Match features – of course, there are no guarantees that same features will be detected and that all matches will be correct

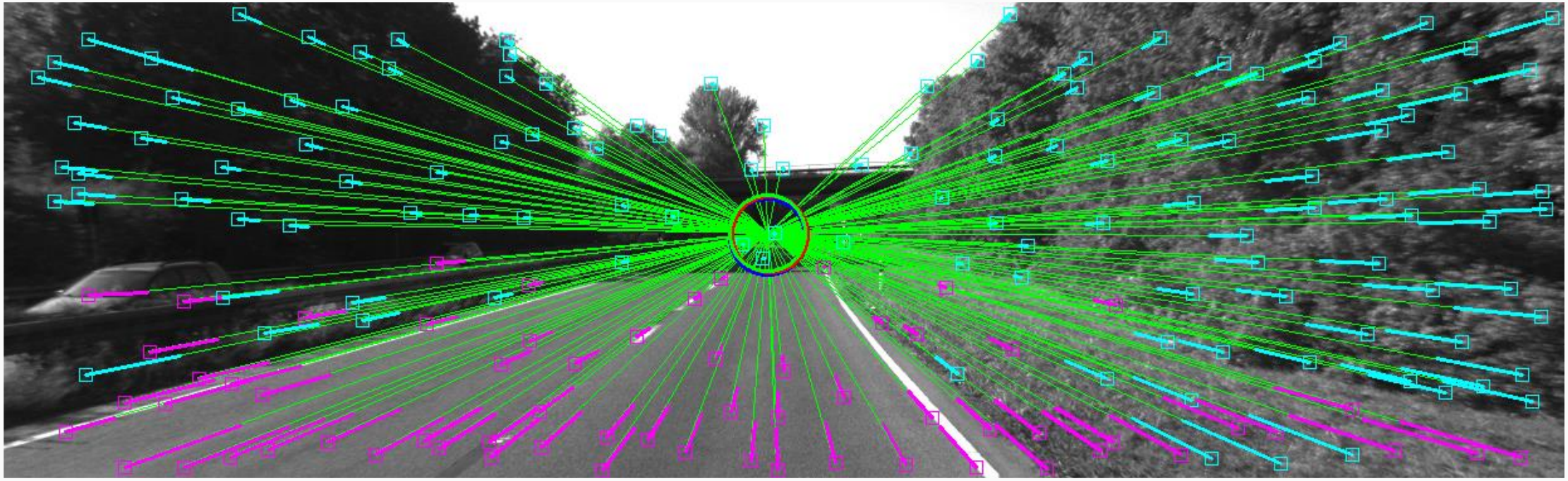
Building blocks of visual odometry



In the first part of the class we talked about camera's principal point, epipoles, epipolar lines, optical flow etc.

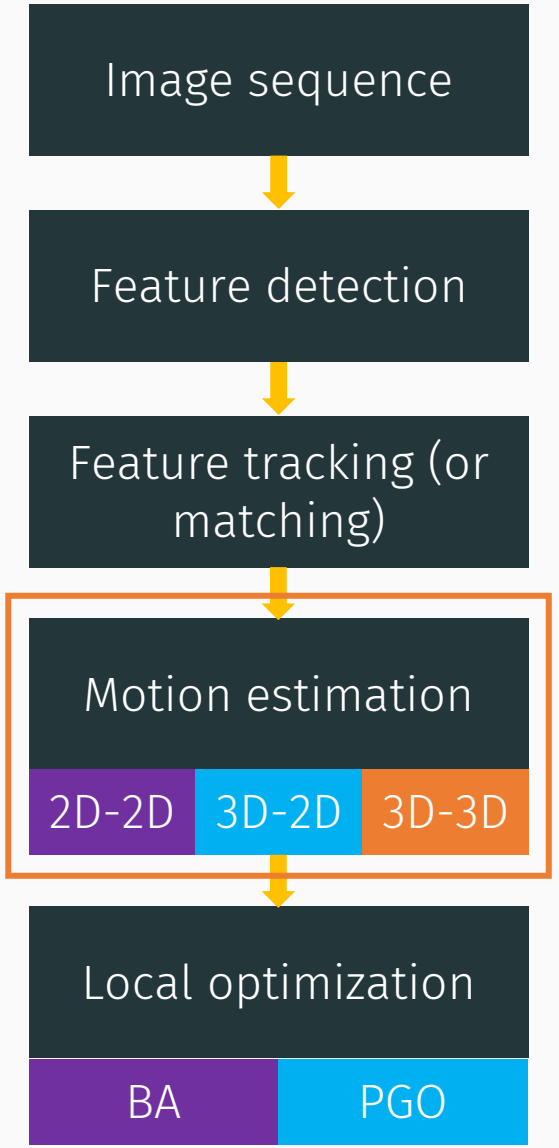
The image (courtesy of SOFT2) shows all the stated objects illustrated with respect to a previuos image taken from a driving car. Can you guess which are which?

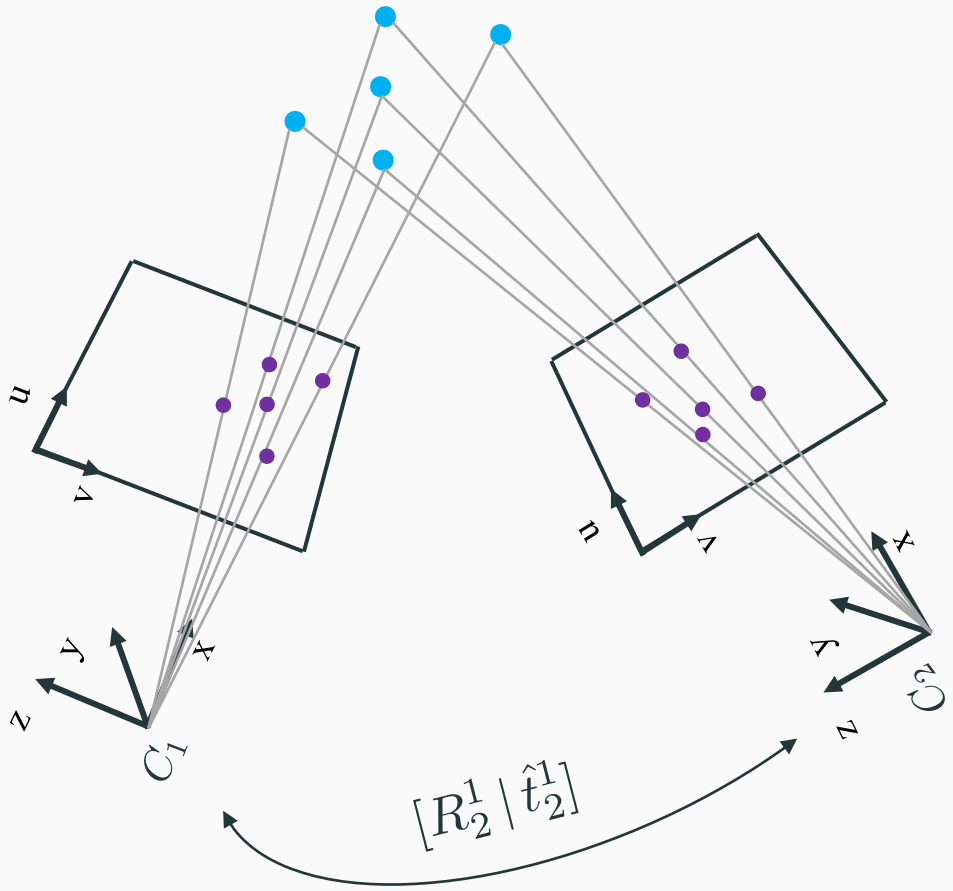
Building blocks of visual odometry



Center of the red circle – camera principle point; centers of the blue and gren circles – epipoles of the previous and current image, respectively (don’t worry, you cannot differentiate the three just based on the image); green lines – epipolar lines w.r.t. the previous image; squares with tails – detected features matched with previous frame features.

Building blocks of visual odometry

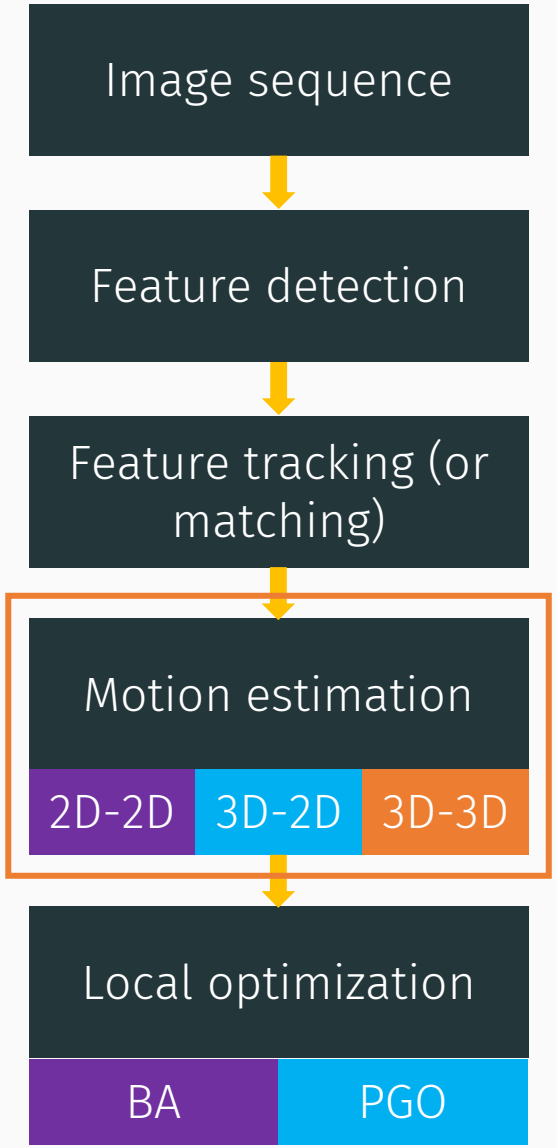


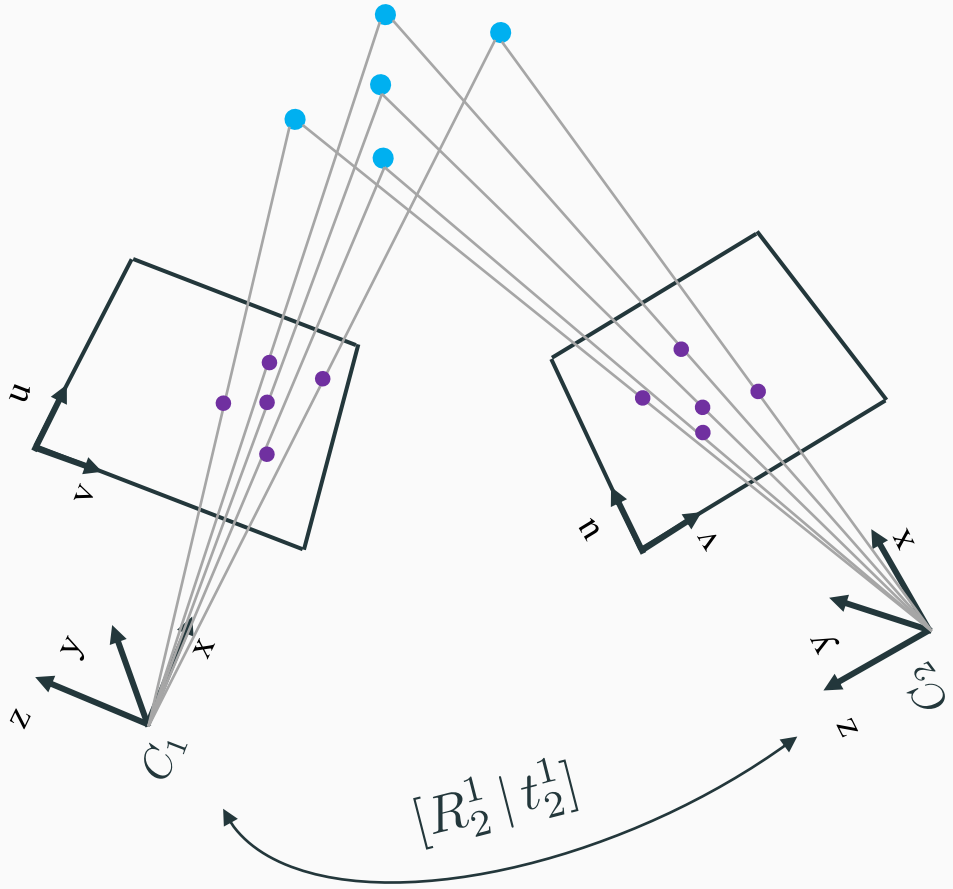


2D-2D: uses matched feature 2D image coordinates from two views (8-pt or 5-pt algorithm -> E matrix -> $[R \mid \hat{t}]$). Note translation is only up to scale with two mono images!

3D-2D: triangulates (only up to scale if mono) matched features from the second view and finds $[R \mid \hat{t}]$ that minimize the distance between the matched 2D image and reprojected 3D features.

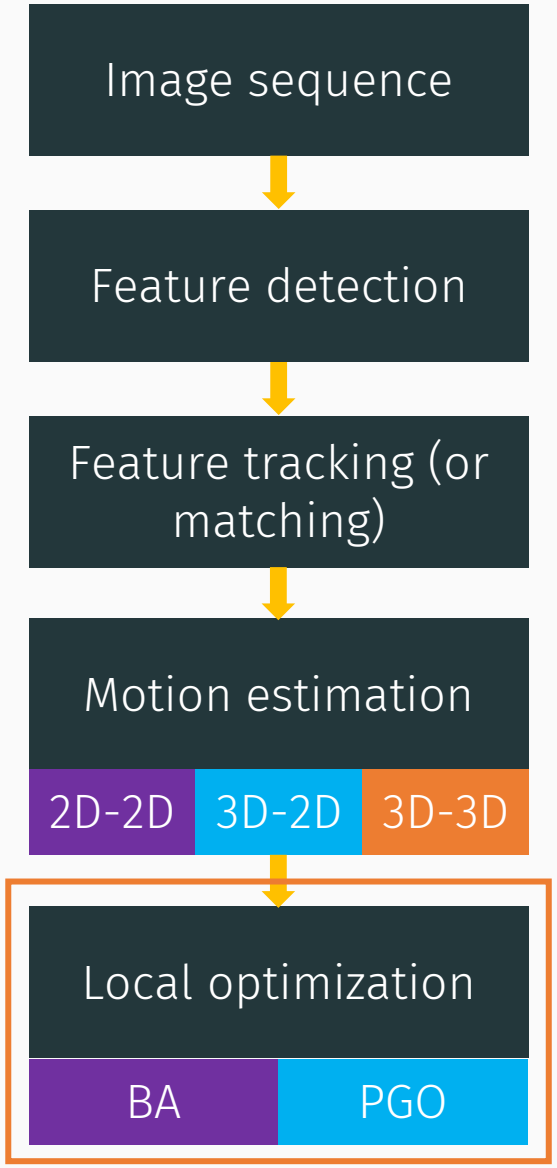
Building blocks of visual odometry

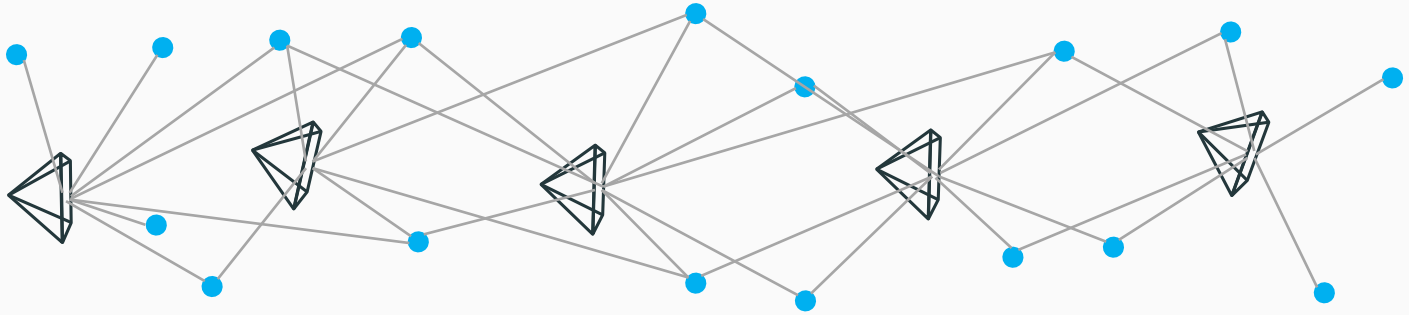




3D-3D: uses two 3D coordinate matched feature sets (point clouds) obtained from two views (via stereo or RGB-D camera) and using, e.g., ICP computes the $[R \mid t]$ (now with scale!).

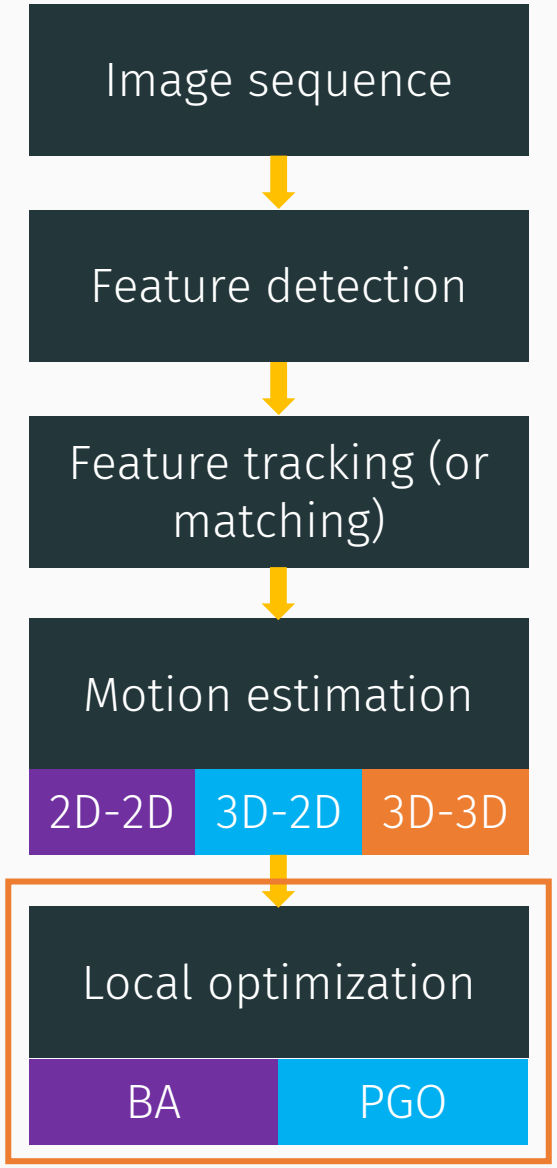
Building blocks of visual odometry

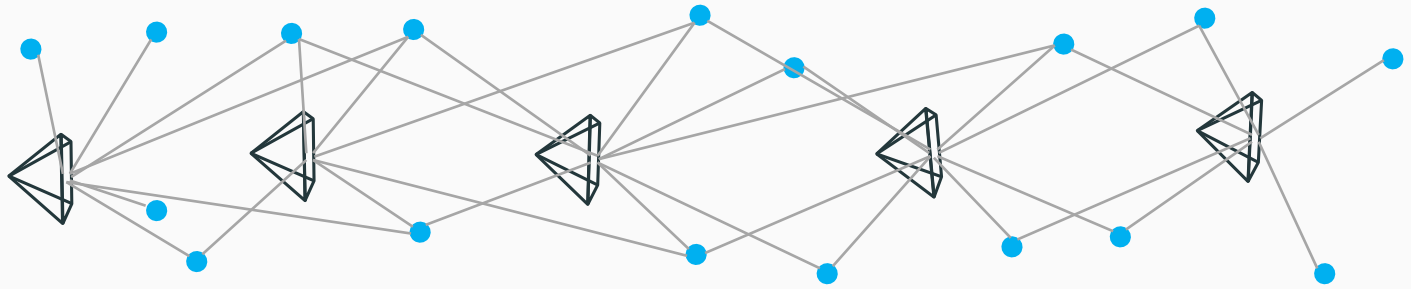




We run visual odometry as usual, by computing relative displacements between subsequent camera pairs, but after each motion estimation, we also take the past *N* (3-5 in practice) frames and run optimization on this *bundle* of frames and we get...

Building blocks of visual odometry

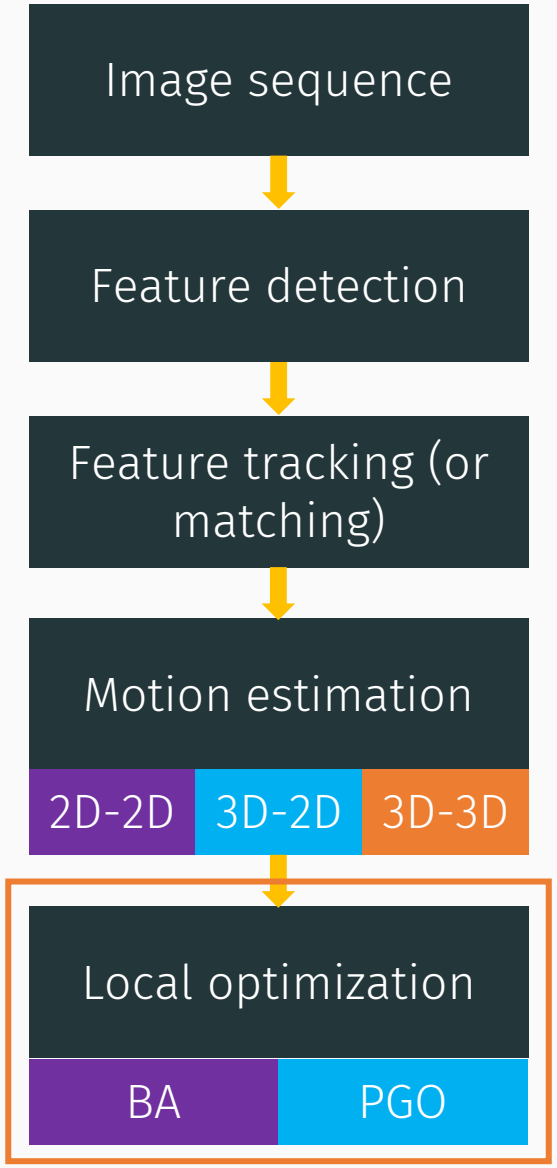


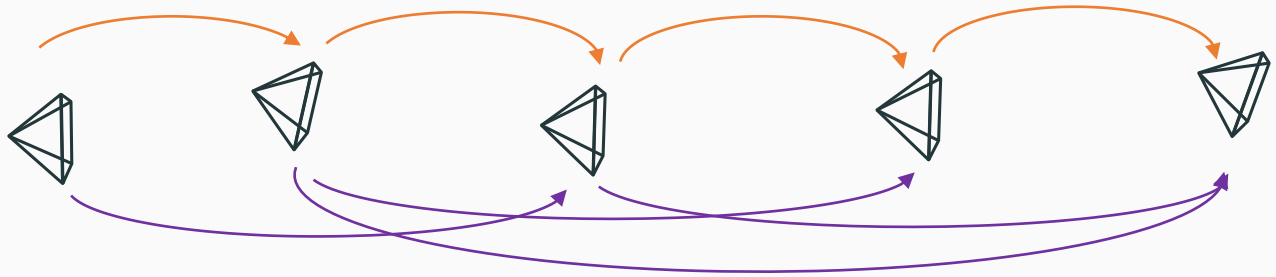


...

Bundle adjustement (BA) - the optimization *adjusts* the poses of the camera and the positions of the features so that a certain error over the whole bundle is minimized (e.g., 3D-2D approach but over multipe frames).

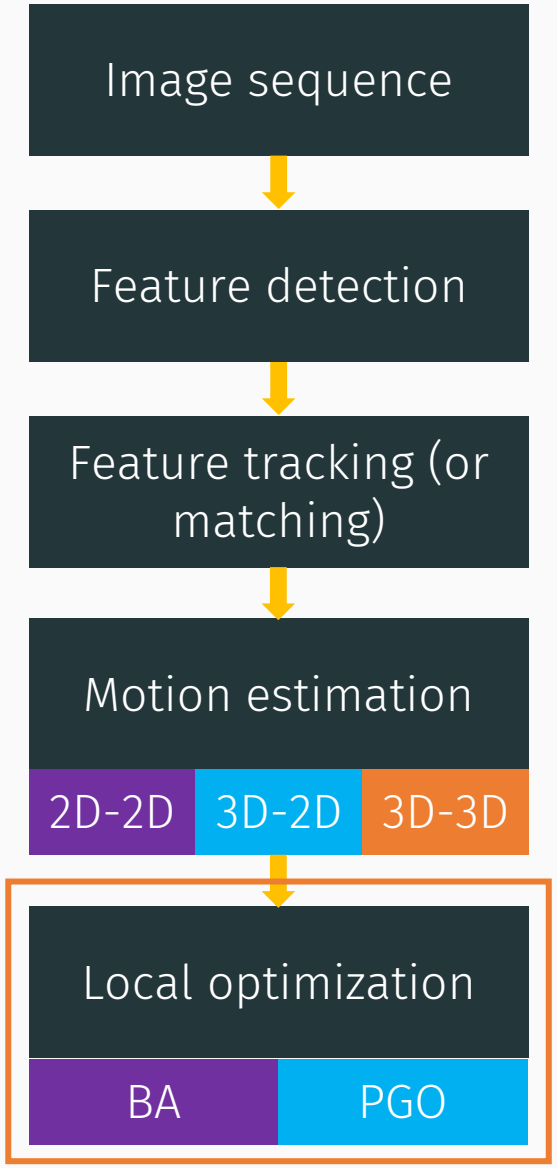
Building blocks of visual odometry

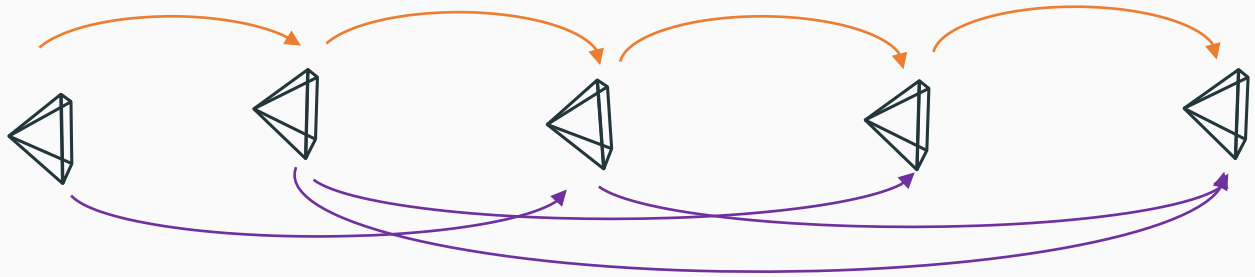




We run visual odometry as usual by computing relative displacements between subsequent camera pairs (orange). Since some feaures are visible across multiple frames, we can use this to compute additional relative displacements (purple, e.g., 1-3, 2-4, 2-5, ...). We can take these multiple motion constraints and we get...

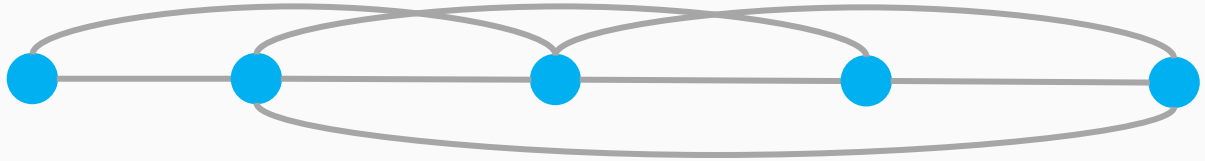
Building blocks of visual odometry





…

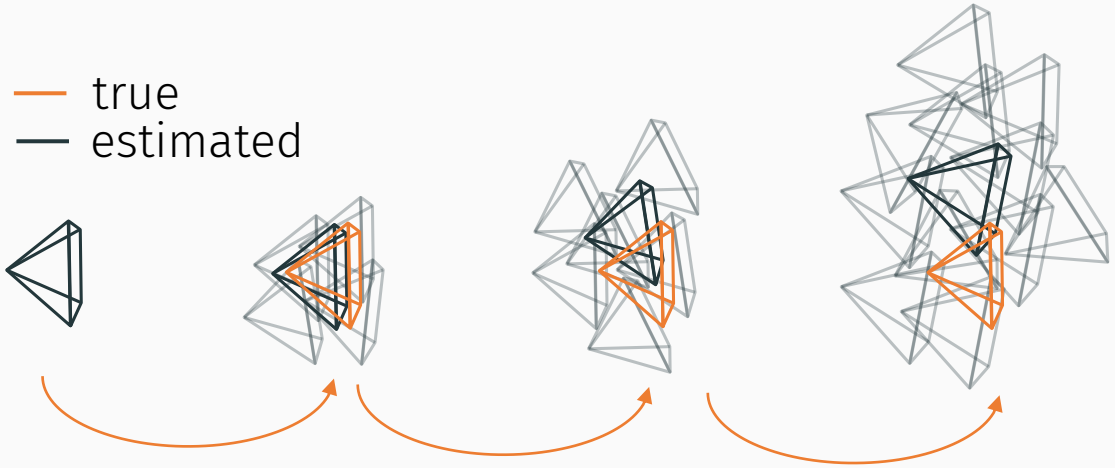
Pose graph optimization (PGO) – we optimize over the set of constraints over a graph of camera poses (nodes are poses, edges are relative displacements) by enforcing local consistency (e.g., 1-2 + 2-3 = 1-3).



VO drift

Estimating ego-motion via odometry is based on concatenating small pose displacements. As such any odometry, whether it is based on wheels, IMU, camera or laser, will be prone to drift.

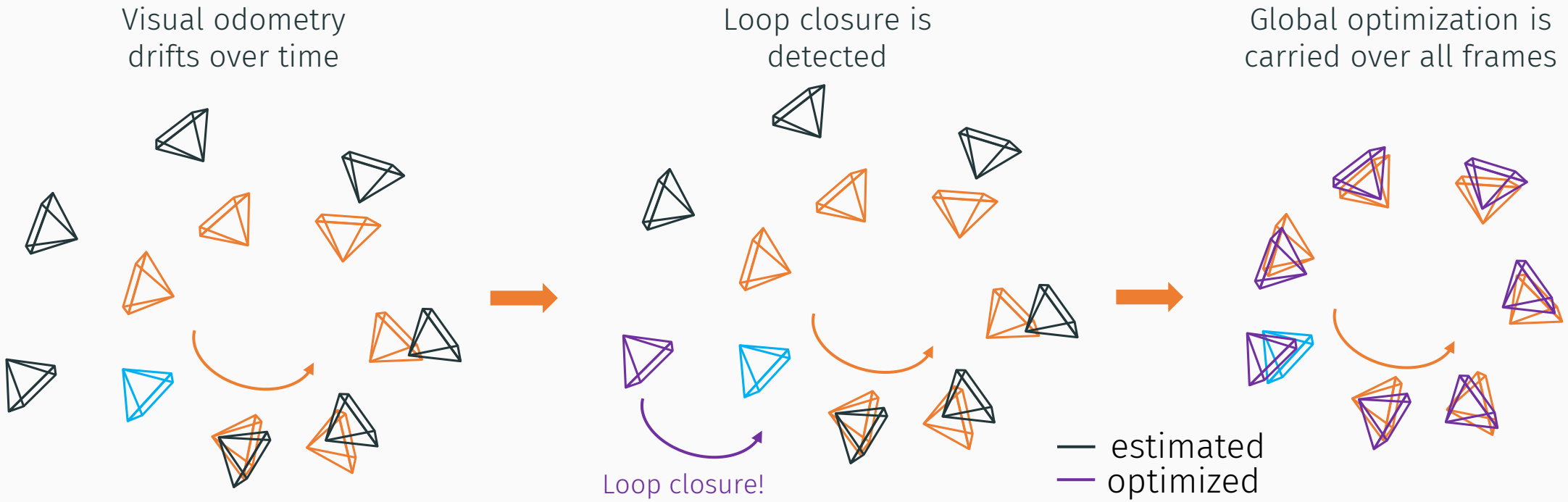
I.e., small displacement errors integrated (summed up) over time can yield rather large final pose errors, thus the uncertainty of each subsequent pose increases.



VO(-SLAM) vs. SfM

To account for the drift, global consistency on the sensor trajectory must be enforced. However, this is only possible if the sensor revisits an already seen part of the environment.

In robotics, this is called loop closing and involves recognizing from an image that this part of the environment has been seen before – place recognition.

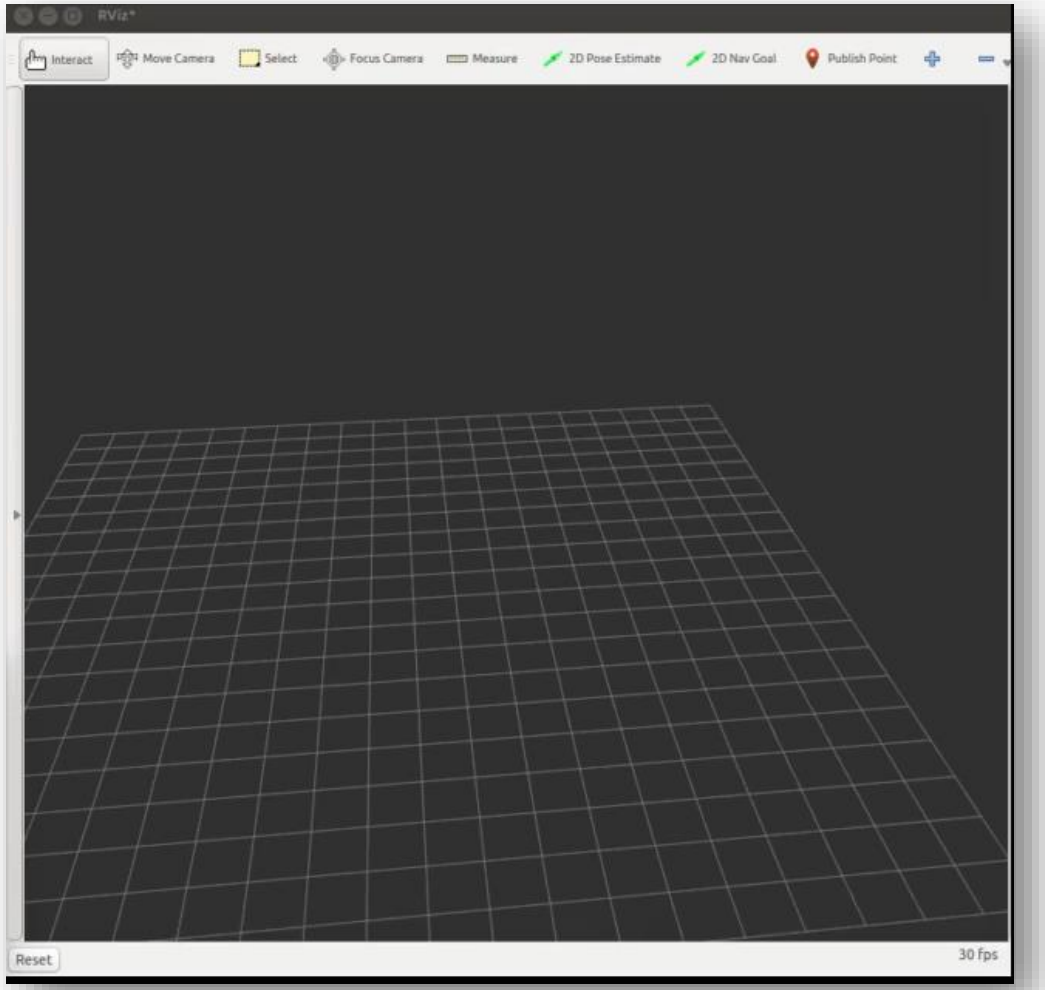


VO(-SLAM) vs. SfM

Visual odometry when coupled with place recognition and optimization that enforces global trajectory consistency is called in robotics literature simultaneous localization and mapping (SLAM).

The example before illustrated optimization over poses-only, but SLAM can also optimize over features, thus essentially building a map and localizing within the map at the same time.

SLAM is a fundamental building block of any autonomus robot or vehicle.



VO(-SLAM) vs. SfM

In computer vision literature you will also often find the term Structure from Motion (SfM).

SfM has as its goal to estimate the camera poses as well as the structure (map) of the environment from an unordered sequence of camera images.

Thus, in the context of vision it is a more general problem than visual odometry or SLAM (ordered, sequential images). We might model this as

$$\mathrm{VO} \subseteq \mathrm{SLAM} \subseteq \mathrm{SfM}.$$

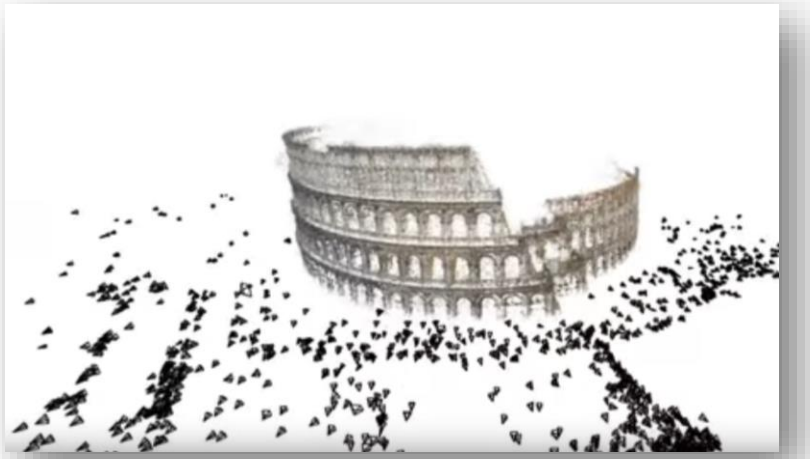
However, note that SLAM is not native to (mono) cameras only, it is a general concept that is applied in robotics to other perception sensors, such as 2D and 3D LiDARs, sonars, radars etc.

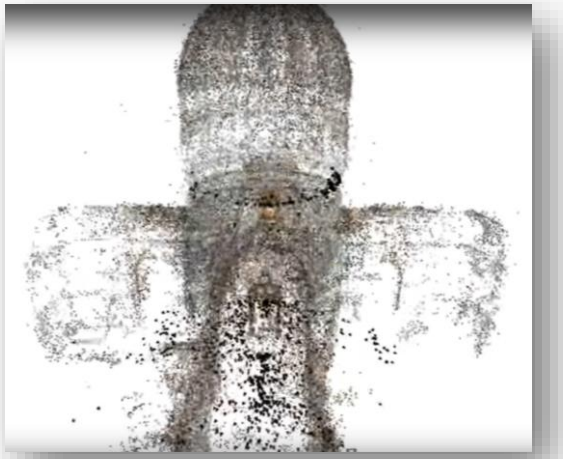
Moreover, by exploiting the fact that sensors are on a vehicle with motion constraints, further simplifications and optimizations can be found.

SfM in action

In 24 hours running on 250 computers the model was constructed from 2,106 Flickr images and 819,242 points. See the paper from 2011.

COLMAP is currently the SfM reference software and can be used as reference solution for other approaches, such as visual odometry and SLAM.





VO(-SLAM) vs. SfM

In conclusion, VO focuses on estimating the sequental motion of the camera (as new frames arrive) and in real time.

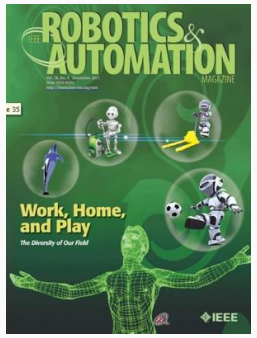
Local bundle adjustment or pose graph optimization can be used (but it's optional) to refine the local estimate of the trajectory in a window.

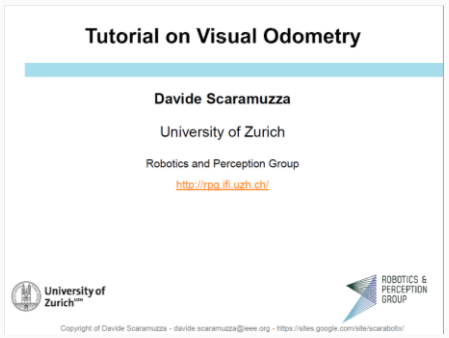
SLAM aims to obtain globally consistent trajectory estimated via loop closure detection. Once loop closure is detected, through global bundle adjustement or pose graph optimization we reduce drift in both the map and/or trajectory.

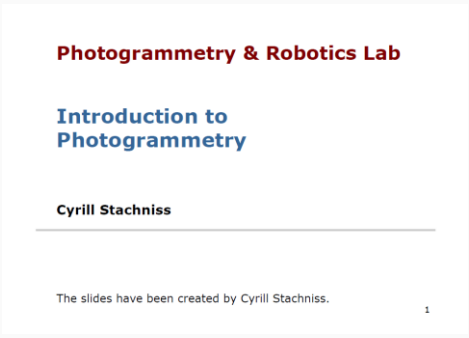
SfM aims to reconstruct globally consistent camera poses and the structure but does not necessarily require the images to be sequentially ordered, like VO and SLAM usually assume.

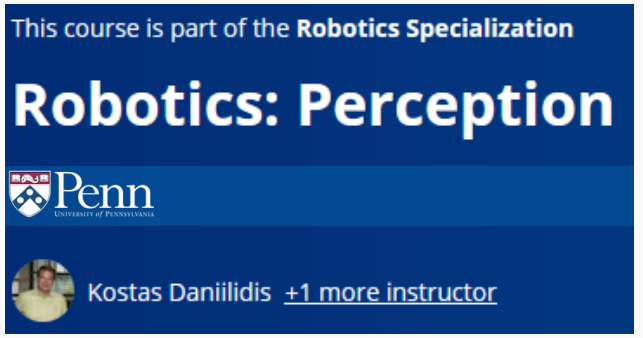
Literature

The slides have been created and inspired by experience, materials from several research papers, tutorials, and similar courses:









- Visual Odometry, Part 1. D. Scaramuzza, F. Fraundorfer, Robotics and Automation Magazine
- Visual Odometry, Part 1. D. F. Fraundorfer, D. Scaramuzza, Robotics and Automation Magazine
- Vision Algorithms for Mobile Robotics, D. Scaramuzza, University of Zurich
- Introduction to Photogrammetry, C. Stachniss, University of Bonn
- Robotics: Perception, K. Daniilidis, J. Shi, University of Pennsylvania

Questions?