
HE VD
IG

Intelligence Artificielle pour les systèmes autonomes (IAA)

Modular pipeline: Odometry, slam and localization

Prof. Yann Thoma - Prof. Marina Zapater

Février 2024

Basé sur le cours du Prof. A. Geiger

Summary

Today's lesson

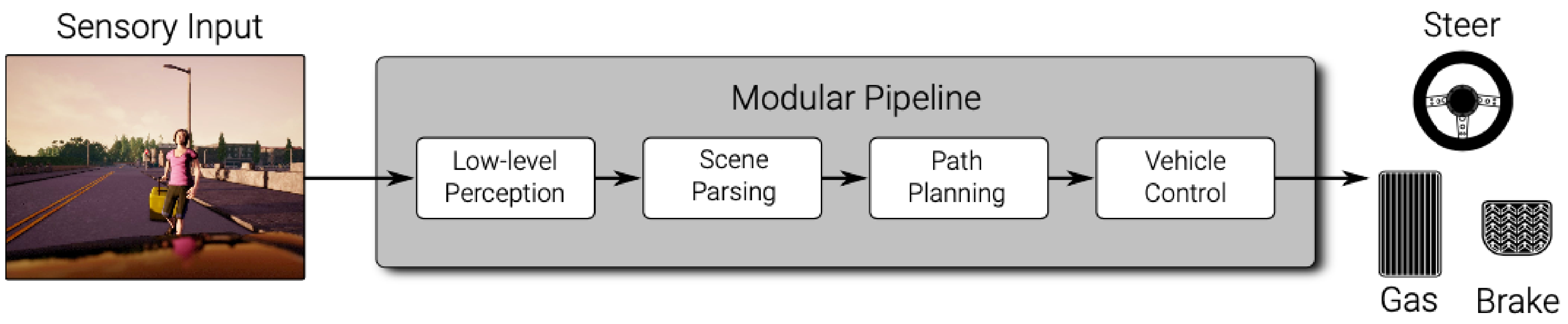
→ **Odometry**

→ SLAM

→ Global localization

Modular Pipeline

Reminder of main blocks



→ Vehicle control (last week's lecture) (chapter 05)

→ **Low-level perception** (today + next lecture)

- Odometry, motion estimation and localization (SLAM) and global localization methods

→ Scene Parsing

→ Path planning

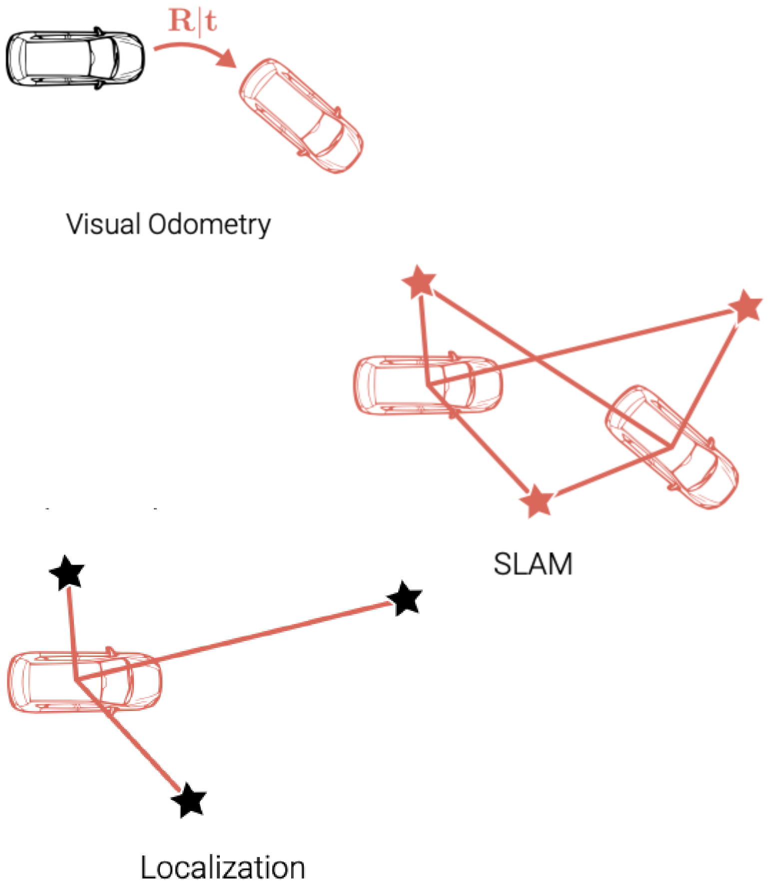
Odometry, motion estimation and localization

Estimation of ego-motion (own motion) and motion of other traffic participants

→ Visual odometry: estimate relative own motion from images

→ SLAM (**S**imulataneous **L**ocalization **A**nd **M**apping) algorithms: building a map and simultaneously localize ourselves on that map

→ Localization methods: finding global positioning of a vehicle given a map



Odometry

Using sensors to estimate changes in own motion over time

→ Odometry provides relative motion estimates
- No global position wrt a map

→ Sensitive to error accumulation over time

→ Can be obtained using a wide variety of sensors
- **Wheel odometry**: wheen encoders to measure wheel rotation
- **Inertial Measurement Units (IMU)** measure a body's forces (acceleration)
- **Visual odometry** : use camera images

A combination of sensors is most often used

Wheel Odometry

Inertial Measurement Unit (IMU)


Visual Odometry

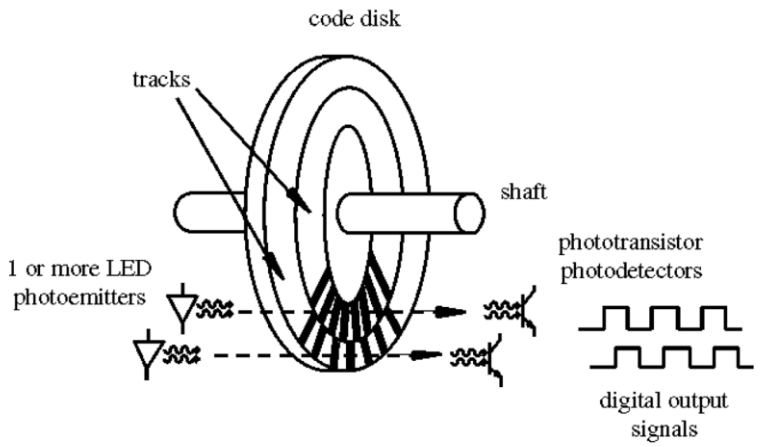
Wheel odometry

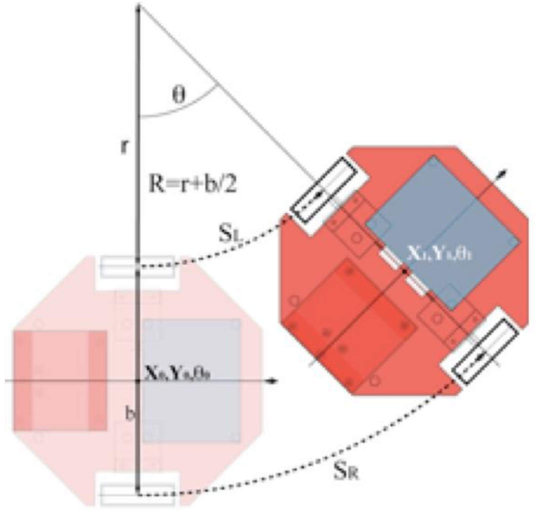
“The easiest” and most often used in robots

→ Using wheel sensors to update position
- Incremental encoders : as wheels rotate, output of photodetectors oscillates between low ang high. We count the number of pulses

→ Using trigonometry to convert number of pulses counted to actual wheel displacement

→ Error sources:
- Limited resolution and pulse-counting errors
- Unequal (or inaccurate measurement) of wheel diameter
- Variation of the contact point of the wheel
- Unequal floor contact, variable friction and slipping





https://www.hmc.edu/lair/ARW/ARW-Lecture01-Odometry.pdf

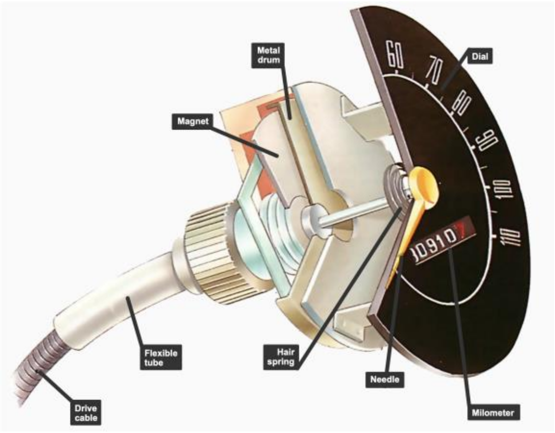
And in a modern car?

Speed meter of a modern car

→ Speed sensor connected to the car's transmission

→ Either mechanical or ferromagnetic

→ After the gear box



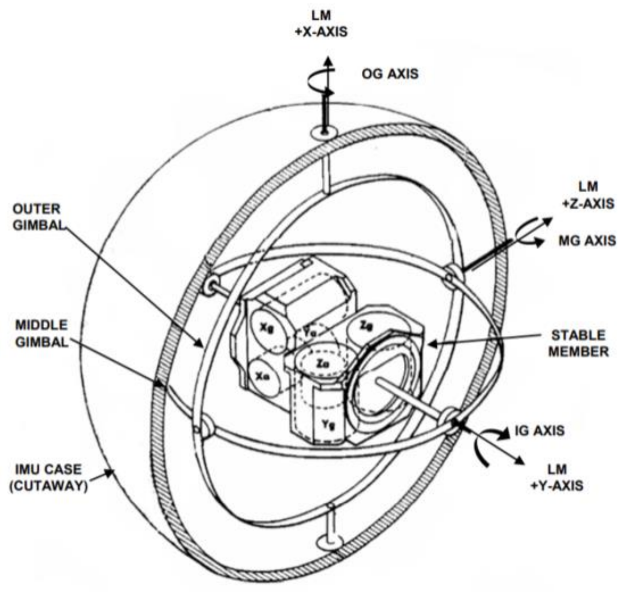
Inertial Measurement Units (IMUs)

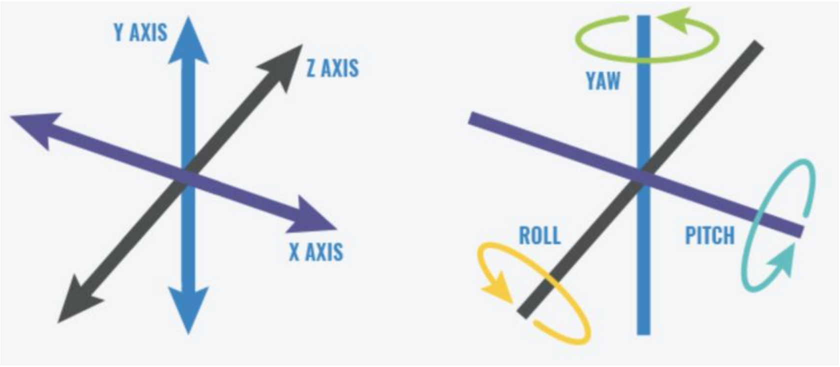
Measuring gravity and angular rate

→ Gyroscopes: provide a measure of angular rate
- Twisting or rotational movement
- Yaw, pitch and roll
- Can be used to detect object orientation in space

→ Accelerometers: measure forces/acceleration
- Measures linear acceleration (changes of velocity) in a single axis
- Normally IMUs are equipped with a 3-axis accelerometer

→ Magnetometer (optional)
- To know the earth's magnetic North
- Determines heading





Visual odometry (VO)

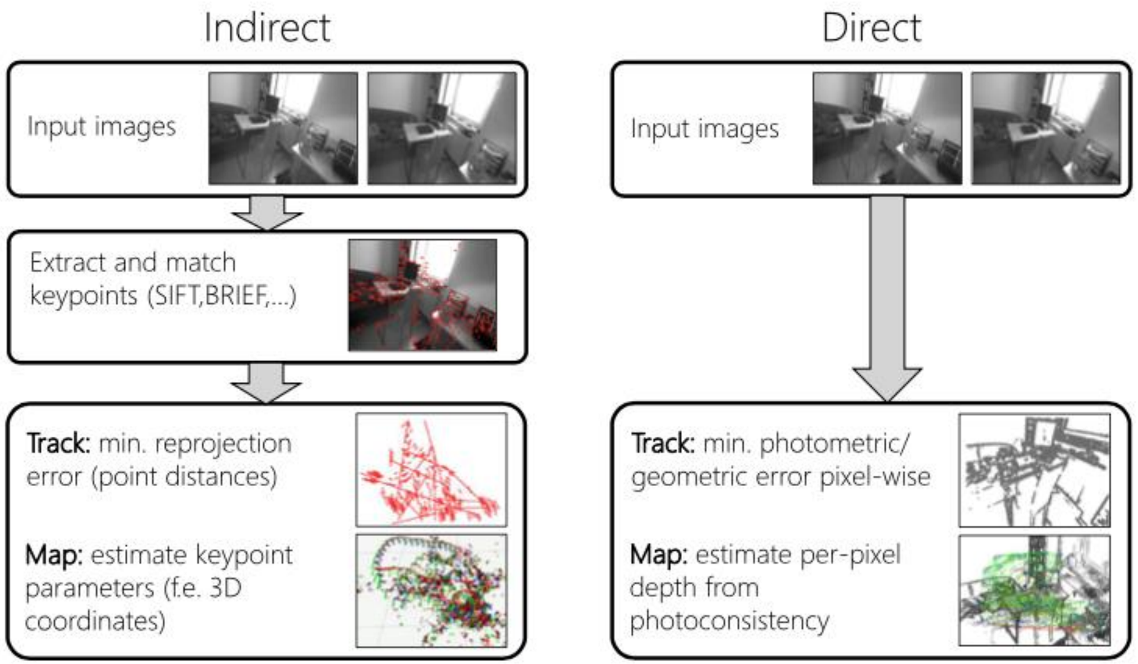
Tracking position and orientation

→ Tracks pose (position **and** orientation) of the camera wrt the environment from images

→ Considers a limited set of recent images: for real-time and to reduce error accumulation

→ A sparse local map is built as a by-product (but is not the main goal)

→ Either "Indirect" of "Direct" methods can be used



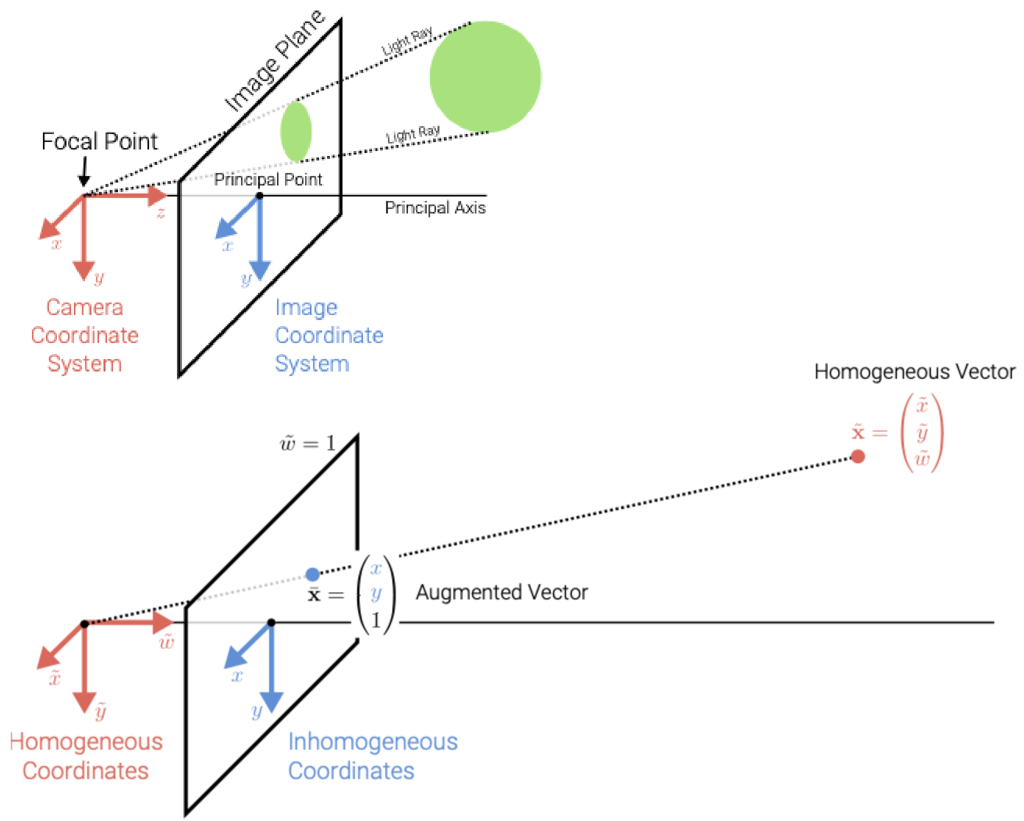
Indirect visual odometry

Extract and match keypoints

→ Detect salient points in the images : edges and blobs
  - A blob is a region in which some properties are similar

→ Match features between two images by their similarity (find correspondences)

→ Requires:
  - Taking perspective into consideration: mapping 2D points into 3D space



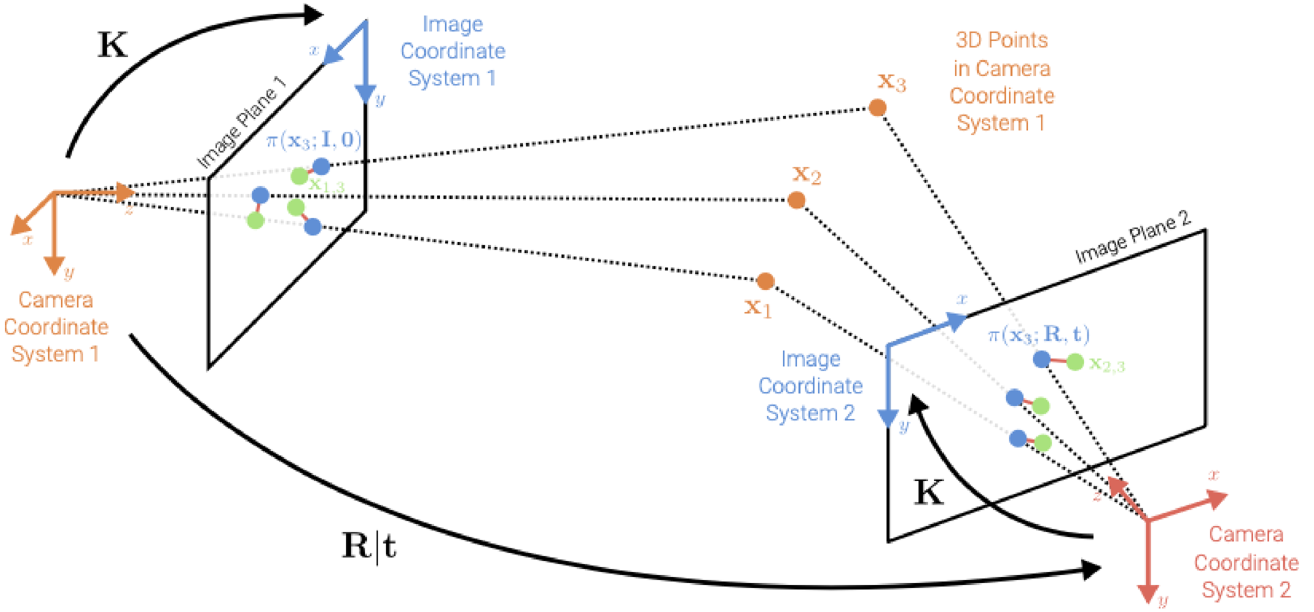
Indirect visual odometry

Extract and match keypoints

→ Detect salient points in the images : edges and blobs
  - A blob is a region in which some properties are similar

→ Match features between two images by their similarity (find correspondences)

→ Requires:
  - Taking perspective into consideration: mapping 2D points into 3D space



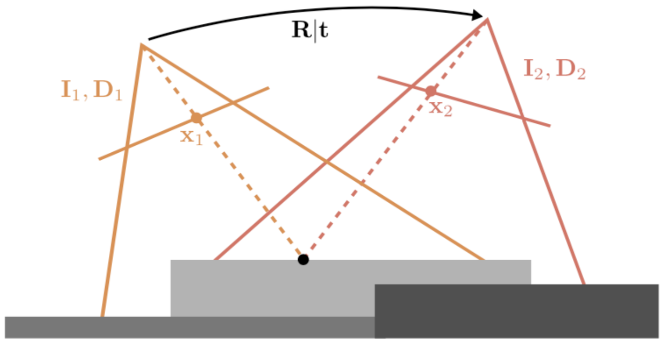
Direct visual odometry

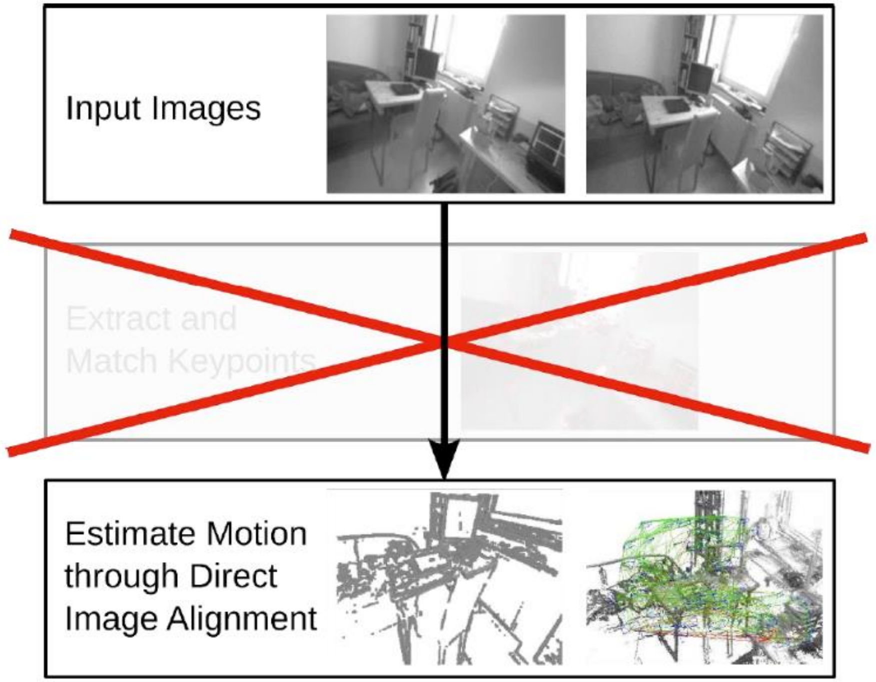
Avoiding keypoint detection and matching

→ Direct image alignment based on pixels

- If we know per-pixel depth, we can simulate an image from a different viewpoint

→ Depth from sensor (lidar, for example) or through multi-view stereo cameras





Summary

Today’s lesson

→ Odometry

→ **SLAM**

→ Global localization

Simultaneous localization and mapping (SLAM)

SLAM: joint optimization of poses and map

→ Until now: considering of 2 adjacent frames, no focus on map

→ Now: considering larger windows → we want poses and location on map

→ SLAM is chicken-and-egg problem: localization requires mapping and vice-versa

→ SLAM aims to correct accumulation errors via loop-closure detection

→ The resulting map is used for localization

Two different formulations for SLAM

Full SLAM vs. online SLAM

→ Full SLAM:

- Estimating the whole trajectory of vehicle and map given all the inputs and measurements
- Very complex! Problem grows very fast

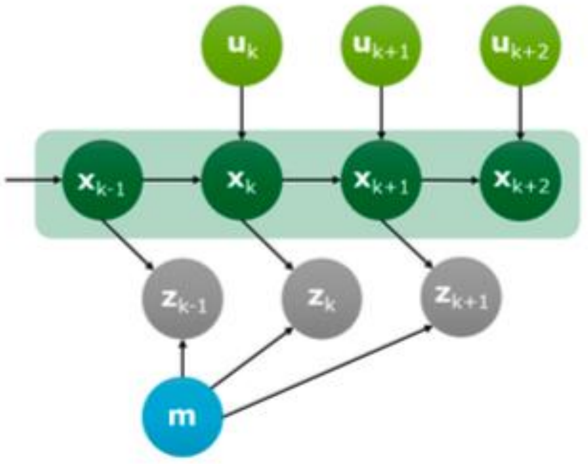
→ Online SLAM (most common today):

- Estimating current position only based on the last sensor information (incremental way)

→ Two common approaches (roughly speaking):

- Filter-based approaches : for online SLAM
- Optimization methods : traditionally used for full SLAM, now used for online SLAM too

Full SLAM



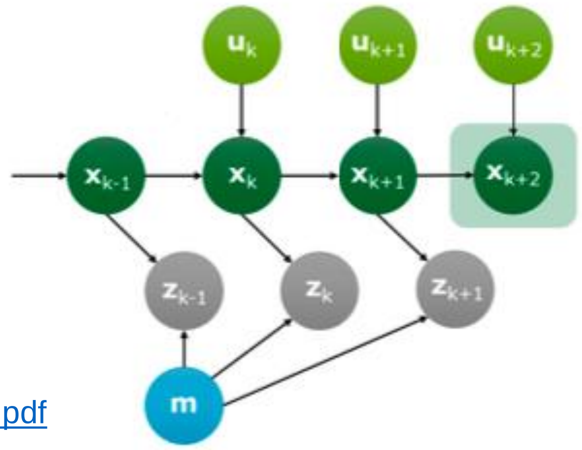
motion estimation

vehicle position

Sensor reading

map

Online SLAM



https://hal.science/hal-01615897/file/2017-simultaneous_localization_and_mapping_a_survey_of_current_trends_in_autonomous_driving.pdf

Common approaches for online SLAM

Filter-based and optimization-based methods

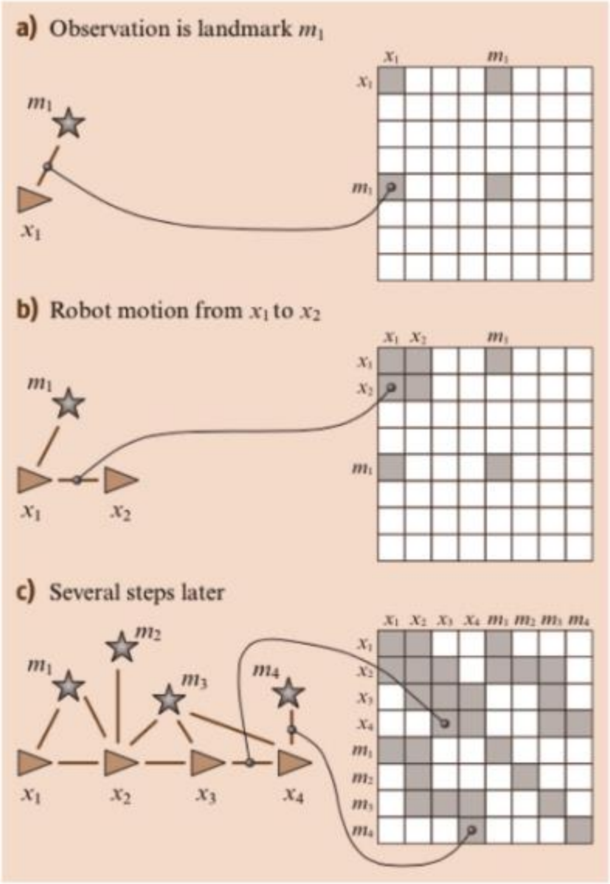
→ Filter-based methods

- Can be used on LIDAR, radar but also in 3D points (vision sensors)
- Example : Extended Kalman Filters - using derivatives of KFs to process data coming from sensors

→ Optimization-based

- Bundle adjustment : jointly optimize 3D structure and the camera parameters
- Graph SLAM: trying to graphically represent the online SLAM problem, and then solve the optimization problem of traversing a graph

Graph SLAM



An example of SLAM

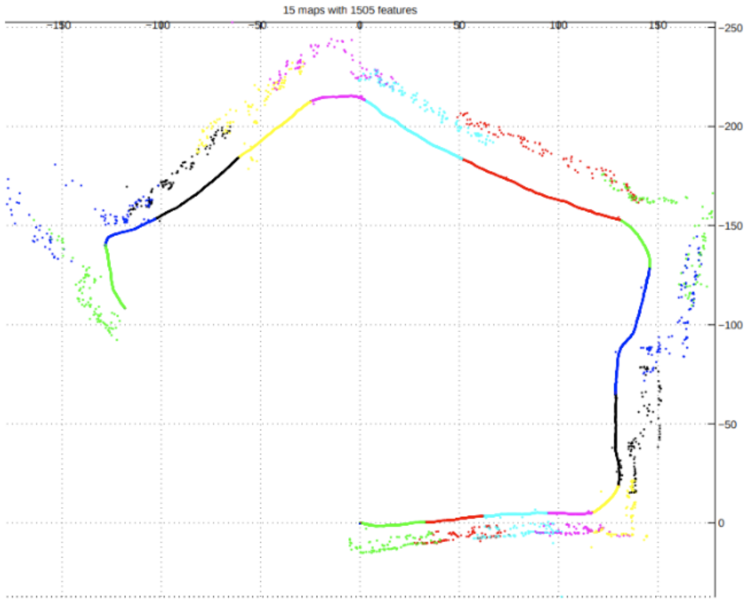
Feature-based SLAM via bundle adjustment

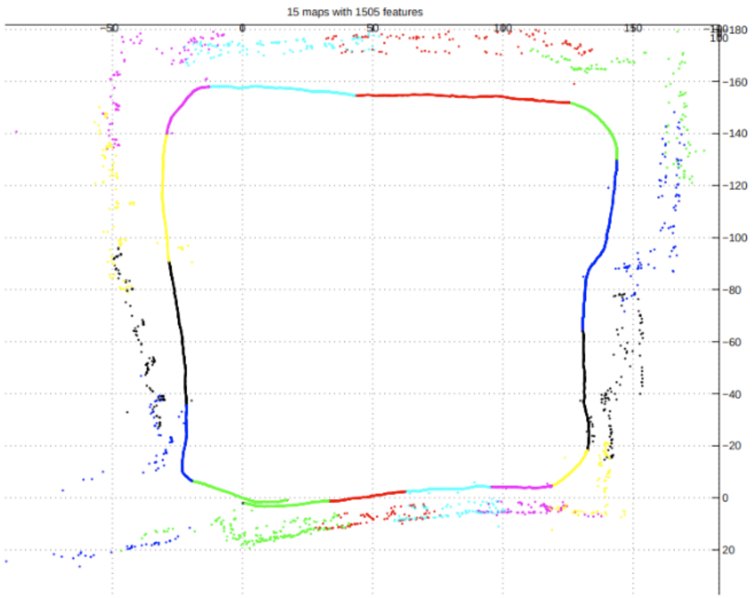
→ Optimizing reprojection errors (distance between observed feature and projected 3D point in image plane) wrt camera parameters and 3D point cloud.

Loop closure detection

Finding correspondences between current and all previous frames

→ And using these correspondences as constraints for optimization





https://www.roboticsproceedings.org/rss03/p38.pdf

Many other algorithms for SLAM

A non-exhaustive table for Indirect SLAM methods

MonoSLAM [1]	PTAM [2]	ORB-SLAM 2 [3]
+ monocular cameras	+ monocular cameras	+ monocular cameras + stereo cameras + RGB-D cameras
- no global consistency	- no global consistency	+ global consistency
Extended Kalman filtering of camera pose and point coordinates. Includes a motion model	Mapping as BA over keyframes, real-time tracking towards keyframe	Mapping as BA over keyframes, real-time tracking towards keyframe, loop-closing via place recognition
Filtering	Bundle adjustment	Bundle adjustment
- local accuracy -- global accuracy	+ local accuracy - global accuracy	+ local accuracy ++ global accuracy

Many other algorithms for SLAM

A non-exhaustive table for Direct SLAM methods

DVO-SLAM [4]	LSD-SLAM [5]	DSO [6]
+ RGB-D cameras	+ monocular cameras + stereo cameras	+ monocular cameras + stereo cameras
+ global consistency	+ global consistency	- no global consistency
camera pose tracking towards keyframe	camera pose tracking towards keyframe	camera pose tracking towards keyframe, camera pose optimization in local keyframe window
+ depth from sensor	+ depth from stereo comparisons & filtering	++ depth optimization in local keyframe window
tracking-only & pose graph optimization	tracking-and-mapping & pose graph optimization	tracking-and-mapping & direct sparse bundle adjustment in local keyframe window with marginalization
+ local accuracy	+ local accuracy	++ local accuracy

Summary

Today’s lesson

→ Odometry

→ SLAM

→ **Global localization**

Vehicle Location

Motivation:

→ Control wrt. a path requires knowledge about the position of the vehicle

→ Sometimes local knowledge is sufficient (lateral position on highway)

→ But often the global location is required (path planning, determining relevant elements that have been marked in a map such as street signs or lane markings)

Localization Approaches:

→ Satellite Localization (uses infrastructure)

→ Visual Localization (uses visual map)

→ Map-based Localization (uses road map)

Satellite Localization

Satellite Systems

- → Galileo: 30 satellites (Europe)
- → GPS: 24 satellites (United States)
- → GLONASS: 24 satellites (Russia)
- → BeiDou-2: 30 satellites (China)

GPS: 4-5 satellites per orbital plane

Satellite Localization

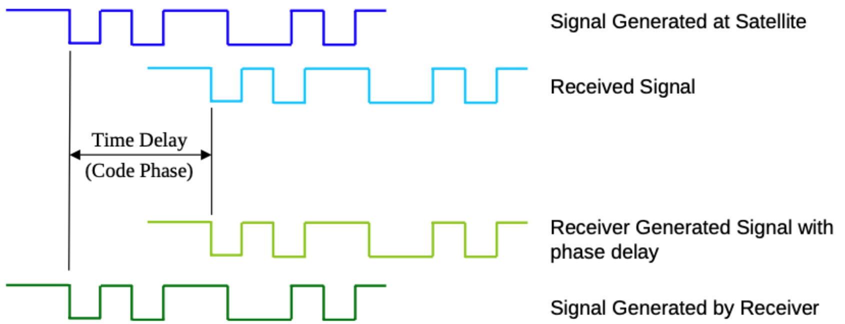
Trilateration

→ Determine location by measuring distance to satellites at known locations

→ Distance to satellite measured as time delay between sent signal/code and received signal/code (receiver knows the code)

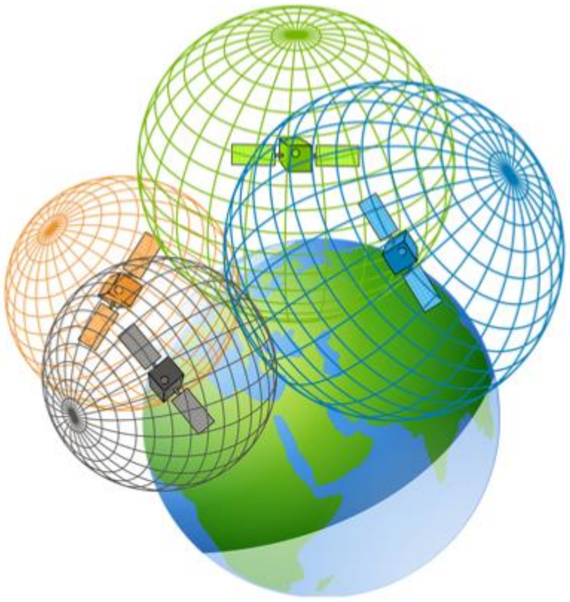
→ Satellites equipped with an onboard atomic clock
- 1 satellite: we know the sphere
- 2 satellites: we know the circle
- 3 satellites: we know the location (2 possibilities)
- 4 satellites: time synchronization and exact location

→ Differential GPS (DGPS):
- Improve measurement using ground stations



Distance = Time Delay * Speed of Light

GPS equations: 4 time measurements and 4 unknowns: $x, y, z, \Delta t$ (see footnote)



http://mason.gmu.edu/~treid5/Math447/GPSEquations/

Satellite Localization

Problems with Satellite Localization

→ Availability

- Satellites not visible in tunnels, narrow city streets
- Dependency on national organizations / interests

→ Accuracy

- 5m-15m for GPS and 0.5m-5m for DGPS
- Only location, no rotation (not full pose)

→ Frequency: max 5-10Hz

→ Sensitive to atmospheric variations and multipath effects (signal reflections)

Visual localization

Main idea

→ Record map/database of known locations with associated features
  - Features can be extracted from images or laser scans

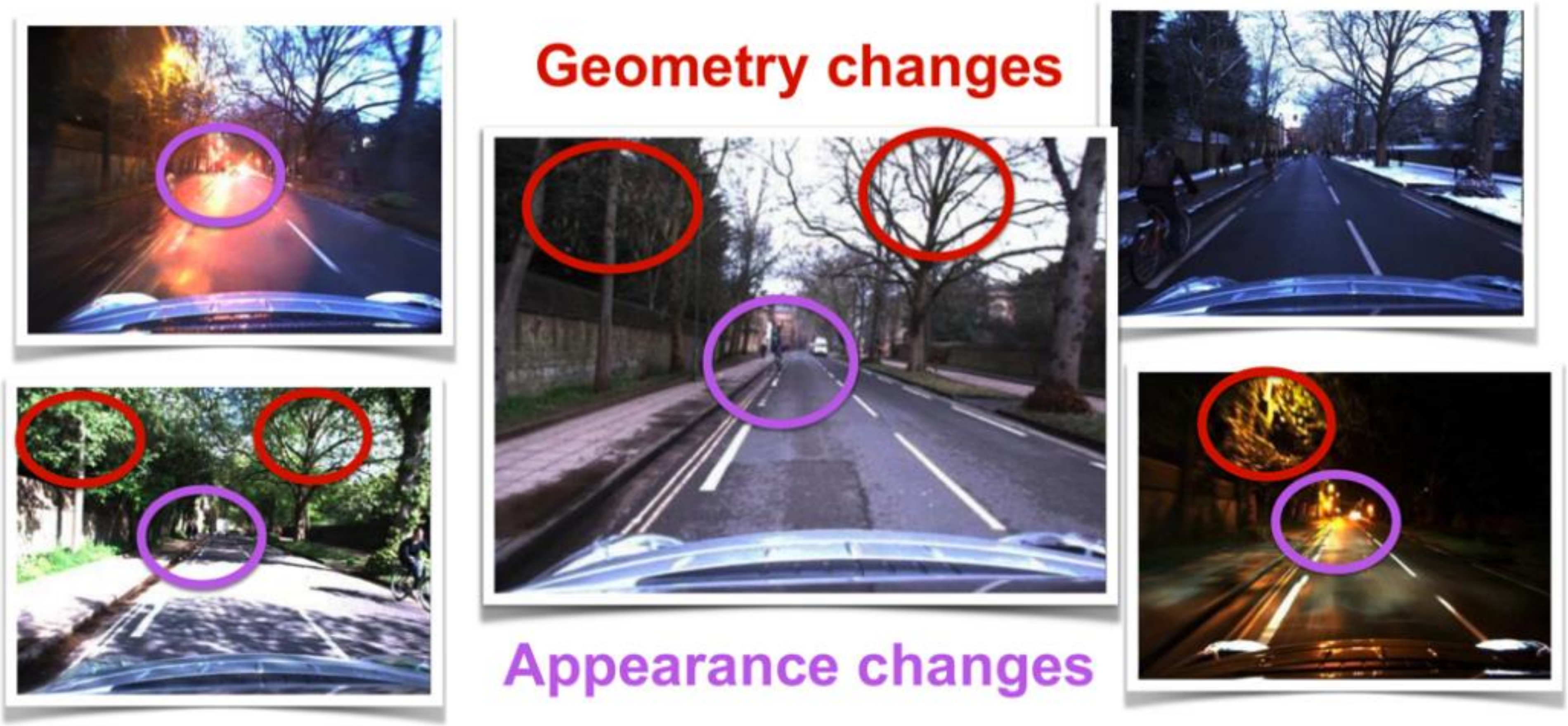
→ Mapping should be conducted under similar conditions as during localization (minimize shift wrt. sensor setup and environmental conditions)

→ At localization time, try to retrieve features extracted from the current input image/scan in the map/database

→ Either localize only at the image level or refine using triangulation or geometric pose estimation

Visual localization

Challenges: same image, but appearance and geometry change…



https://robotcar-dataset.robots.ox.ac.uk/images/robotcar_ijrr.pdf

Visual localization

Topometric localization

→ Setup: Record sequence with cameras, Lidar and GPS as ground truth

→ Goal: Localize new image (different day/season) wrt. recorde sequence

→ Creating a map via a direct graph, where nodes (=frames) are created based on distance threshold.

Visual localization

Learning-based localization

PoseNet:

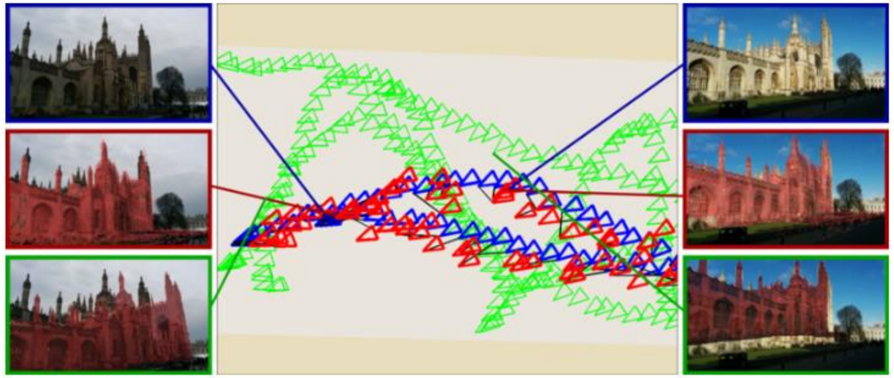
→ Input: Single RGB image

→ Output: 6 DOF camera pose

→ 23 layer deep convolutional network based on GoogleNet

→ More robust than feature-based methods

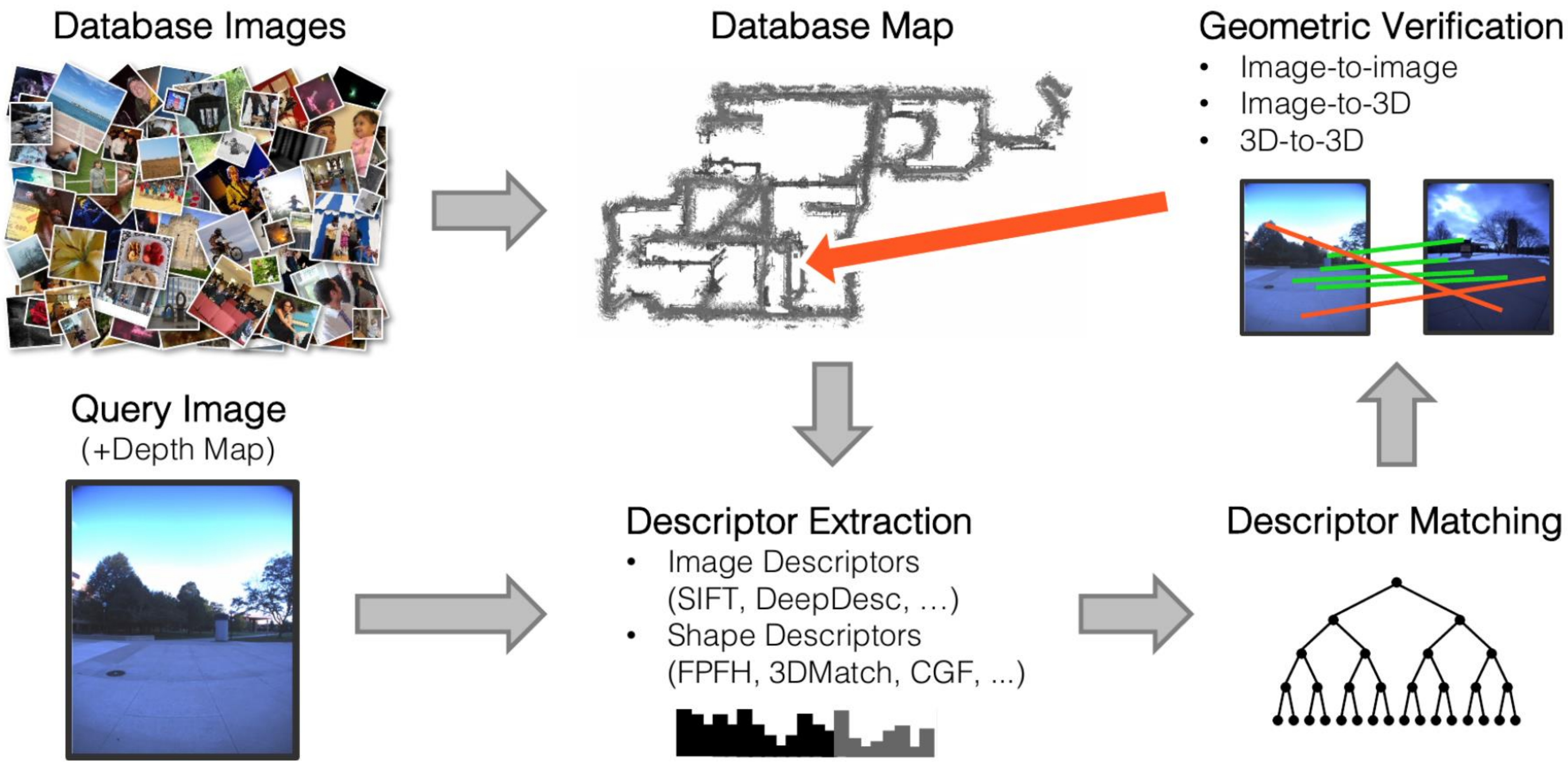
→ Less accurate than feature-based methods



(green = train, red = pred, blue = GT)

Visual localization

Feature-based localization - Overview

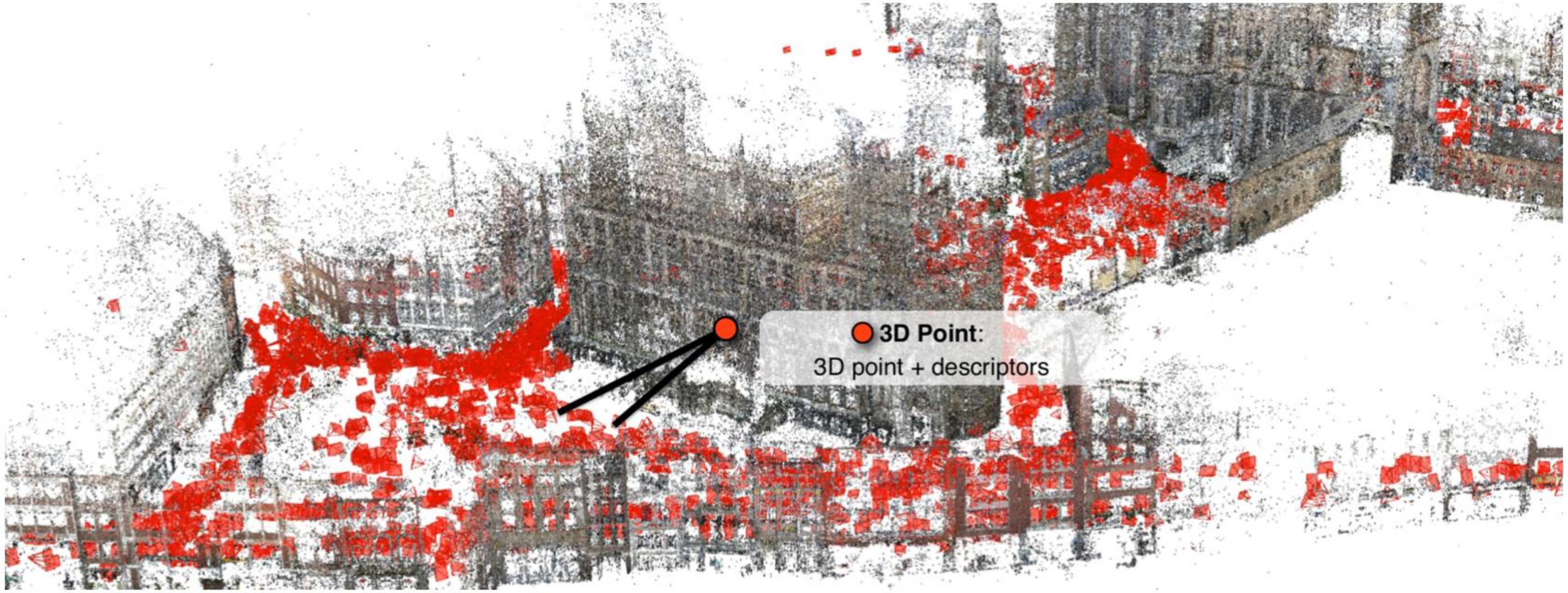


Feature-based localization

Mapping

→ Sparse 3D reconstruction based on sparse feature correspondences (SLAM)

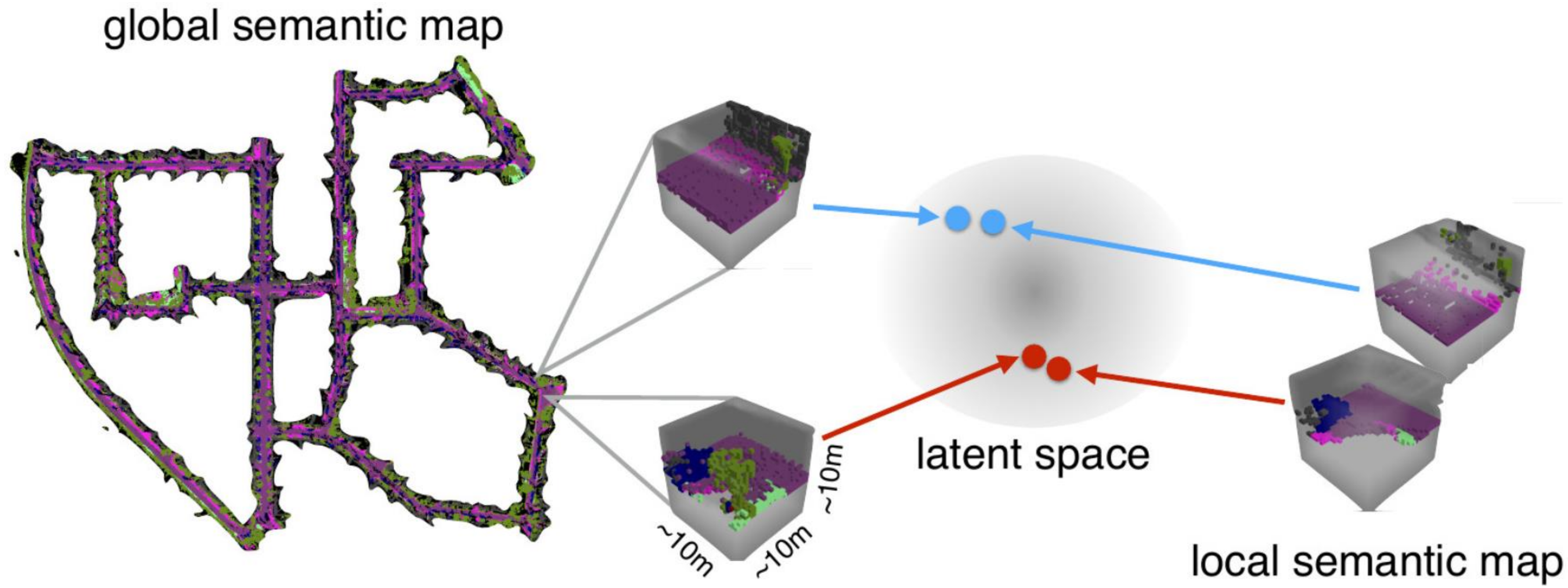
→ Associate each 3D point with a local image descriptor (e.g., SIFT, S



Feature-based localization

Mapping

→ Semantic viewpoint invariant feature representations can facilitate matching



Feature-based localization

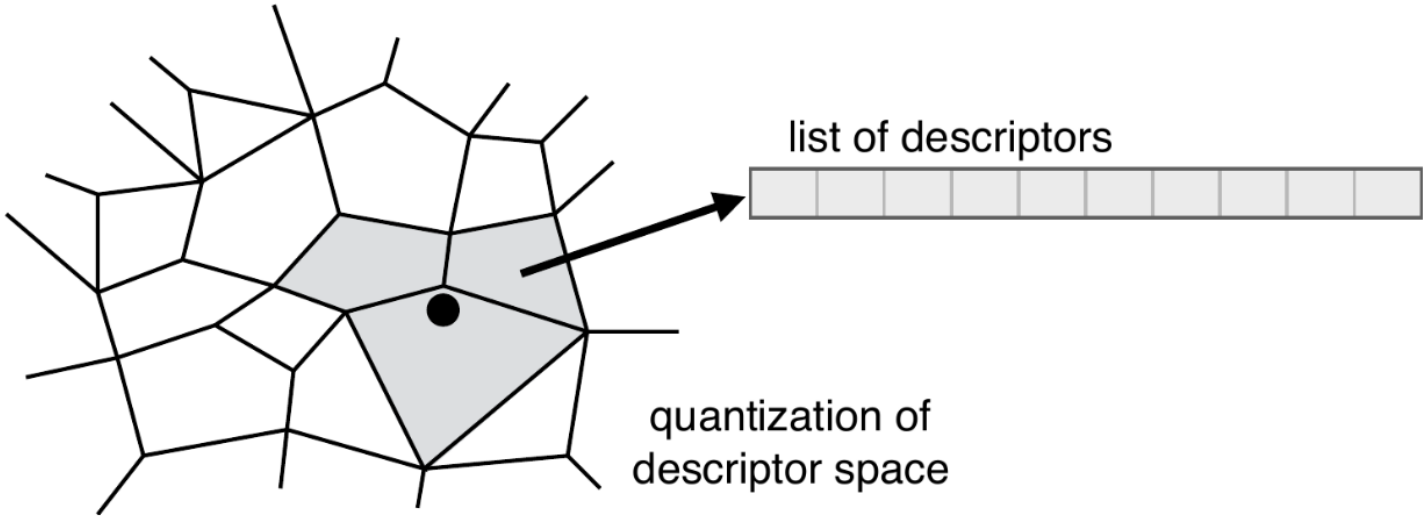
Descriptor matching

→ Search nearest neighbours of features from query image (in descriptor space)

→ Approximate search due to curse of dimensionality

→ Popular approaches: k-dtrees, invertedindex,...

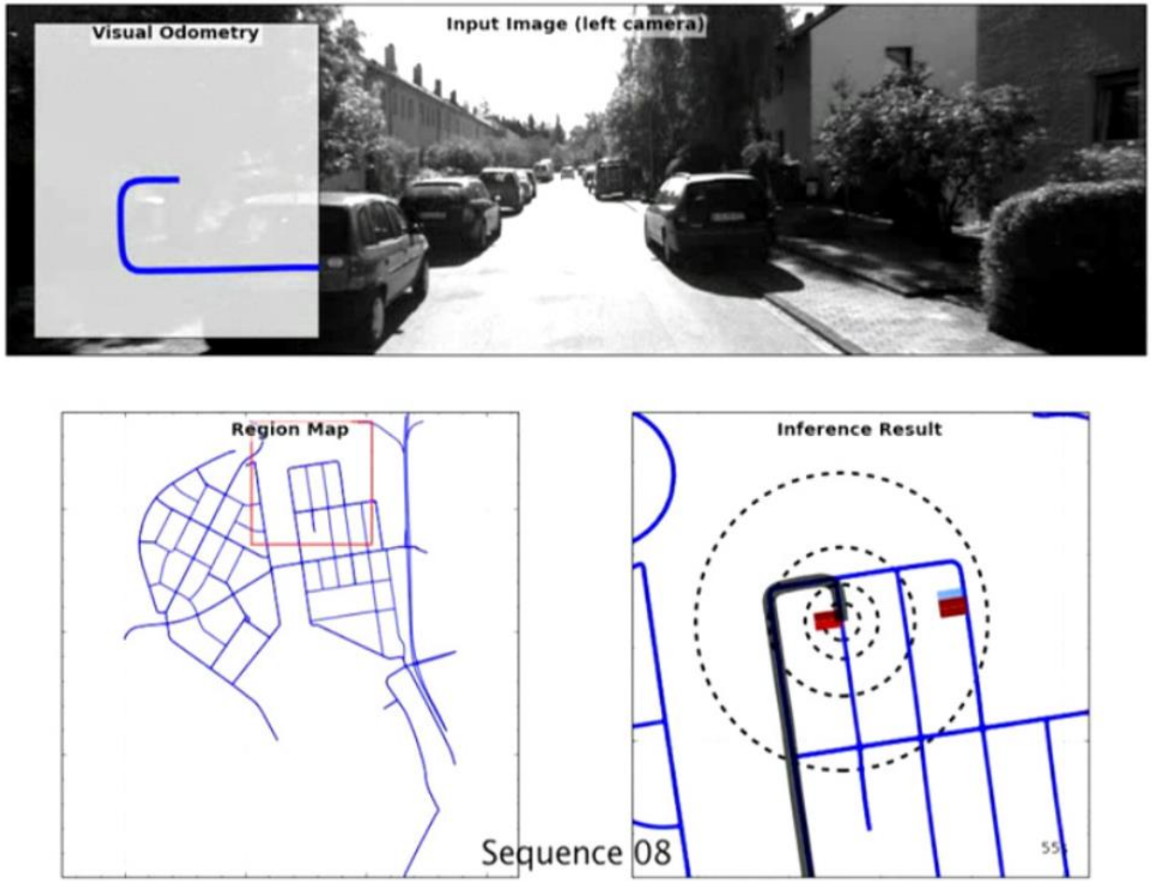
→ Good software packages available: FLANN FAISS



Map-based localization

Using Visual Odometry (VO)

→ Update location probability distribution on map using VO measurements



HE VD
IG

REDS
Institut
Reconfigurable
and Embedded
Digital Systems

Hes·so
Haute Ecole Spécialisée
de Suisse occidentale