Mapping and SLAM

ST5 Autonomous robotics

Francis Colas

2022-09-23

Introduction

Localization

- estimate the pose of the robot in a known map
- Bayesian filters: Markov or Monte-Carlo localization

Introduction

Localization

- estimate the pose of the robot in a known map
- Bayesian filters: Markov or Monte-Carlo localization

Mapping

- build a map based on sensor values
- easy to do with known poses

Introduction

Localization

- estimate the pose of the robot in a known map
- Bayesian filters: Markov or Monte-Carlo localization

Mapping

- build a map based on sensor values
- easy to do with known poses

Simultaneous Localization and Mapping

- jointly solve localization and mapping
- from sensor values

Introduction

Localization

- estimate the pose of the robot in a known map
- Bayesian filters: Markov or Monte-Carlo localization

Mapping

- build a map based on sensor values
- easy to do with known poses

Simultaneous Localization and Mapping

- jointly solve localization and mapping
- from sensor values

Aim of this session

- mapping algorithms
- SLAM algorithms

01

Mapping

Mapping

Definition

- build a map
- from sensor values
- knowing the pose of the robot at all times
- algorithms are different depending on the map
- sensors values and models are important

Mapping

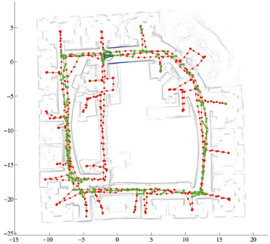
Definition

- build a map
- from sensor values
- knowing the pose of the robot at all times
- algorithms are different depending on the map
- sensors values and models are important

Examples

- pose graph
- point cloud
- occupancy grid

Building of a pose graph



Building of a pose graph

Elements

- node with pose and sensor values
- links between nodes: relative displacement

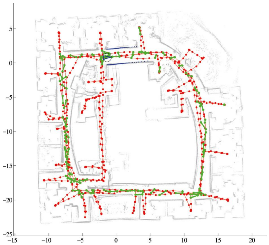
Building of a pose graph

Elements

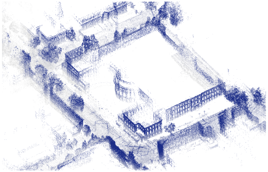
- node with pose and sensor values
- links between nodes: relative displacement

Algorithm

- with each new sensor value:
  - add node to the graph
  - link to previous node thanks to localization
  - eventually look for other close nodes (loop closure)



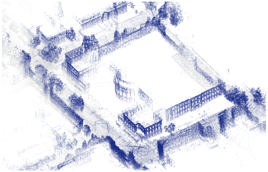
Building of a point cloud



Building of a point cloud

Sensor values

- depth image
- 2D or 3D laser scan
- in sensor reference frame



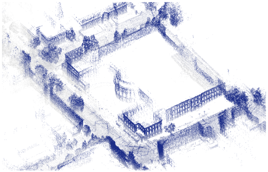
Building of a point cloud

Sensor values

- depth image
- 2D or 3D laser scan
- in sensor reference frame

Algorithm

- with each new sensor value:
  - transform into point cloud in map frame
  - concatenate into point cloud map
- sometimes: clean map



Building of an occupancy grid

Building of an occupancy grid

Sensor values

- 2D laser scan (for 2d occupancy grids)
- 3D scan or depth image (for 3d grids)
- distance measured from center of sensor

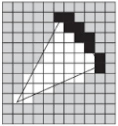
Building of an occupancy grid

Sensor values

- 2D laser scan (for 2d occupancy grids)
- 3D scan or depth image (for 3d grids)
- distance measured from center of sensor

Algorithm

- for each measured distance:
  - cast ray from center of sensor until impact
  - update occupancy values along ray



Ray casting

Occupancy grid update

Ray casting

- start: absolute position of the center of the sensor
- end: absolute position of the impact
- Bresenham's line algorithm[1]

[1] https://en.wikipedia.org/wiki/Bresenham%27s_line_algorithm

Occupancy grid update

Ray casting

- start: absolute position of the center of the sensor
- end: absolute position of the impact
- Bresenham's line algorithm[1]

Cell update

- probabilistic
  - impact position: increase probability of occupancy
  - along ray: decrease probability of occupancy

[1] https://en.wikipedia.org/wiki/Bresenham%27s_line_algorithm

Occupancy grid update

Ray casting

- start: absolute position of the center of the sensor
- end: absolute position of the impact
- Bresenham's line algorithm[1]

Cell update

- probabilistic
  - impact position: increase probability of occupancy
  - along ray: decrease probability of occupancy
- deterministic
  - impact position: set probability to 1
  - along ray: set probability to 0

[1] https://en.wikipedia.org/wiki/Bresenham%27s_line_algorithm

Conclusion on mapping

Map building

- transform sensor-centered values into global frame
- accumulation into a map
- relatively easy when pose is known

Conclusion on mapping

Map building

- transform sensor-centered values into global frame
- accumulation into a map
- relatively easy when pose is known

Conclusion on mapping

Map building

- transform sensor-centered values into global frame
- accumulation into a map
- relatively easy when pose is known

Localization

- estimate the pose
- when the map is known

02

SLAM

SLAM

Definition

- Simultaneous Localization And Mapping

SLAM

Definition

- Simultaneous Localization And Mapping

SLAM

Definition

- Simultaneous Localization And Mapping
- solving both problem jointly
- unknown pose
- unknown map

SLAM

Definition

- Simultaneous Localization And Mapping
- solving both problem jointly
- unknown pose
- unknown map

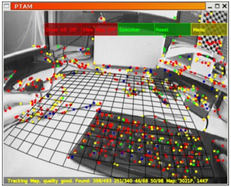
Approaches

- optimization:
  - in parallel: map optimization and pose optimization
- probabilistic
  - state estimation (pose+map) with an EKF
  - state estimation (pose+map) with a particle filter

PTAM

Parallel Tracking and Mapping

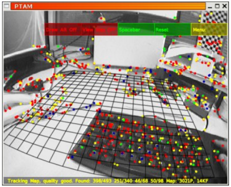
- for augmented reality: estimation of the camera pose
- optimization approach
- no odometry but camera motion model
- landmark map: visual features with descriptors



PTAM

Parallel Tracking and Mapping

- for augmented reality: estimation of the camera pose
- optimization approach
- no odometry but camera motion model
- landmark map: visual features with descriptors



Algorithm

- Tracking:
  - fast (30 Hz) estimate of camera pose from the current map
  - optimization of reprojection error of map features in the image
- Mapping:
  - slow (~ Hz) optimization of the map
  - integration of keyframes by bundle adjustment

Probabilistic approach

Probabilistic approach

- joint estimation of pose and map

$$p(\boldsymbol{x}_k, \boldsymbol{m} \mid \boldsymbol{z}_{1:k}, \boldsymbol{u}_{1:k})$$

Probabilistic approach

Probabilistic approach

- joint estimation of pose and map

$$p(\boldsymbol{x}_k, \boldsymbol{m} \mid \boldsymbol{z}_{1:k}, \boldsymbol{u}_{1:k})$$

- variables
  - pose: $\boldsymbol{x}_k$
  - observation: $\boldsymbol{z}_k$
  - command: $\boldsymbol{u}_k$
  - map: $\boldsymbol{m}$

Probabilistic approach

Probabilistic approach

- joint estimation of pose and map

$$p(\boldsymbol{x}_k, \boldsymbol{m} \mid \boldsymbol{z}_{1:k}, \boldsymbol{u}_{1:k})$$

- variables
  - pose: $\boldsymbol{x}_k$
  - observation: $\boldsymbol{z}_k$
  - command: $\boldsymbol{u}_k$
  - map: $\boldsymbol{m}$
- models
  - motion: $p(\boldsymbol{x}_{k+1} \mid \boldsymbol{x}_k, \boldsymbol{u}_{k+1})$
  - observation: $p(\boldsymbol{z}_k \mid \boldsymbol{x}_k, \boldsymbol{m})$

Probabilistic approach

Probabilistic approach

- joint estimation of pose and map

$$p(\boldsymbol{x}_k, \boldsymbol{m} \mid \boldsymbol{z}_{1:k}, \boldsymbol{u}_{1:k})$$

- variables
  - pose: $\boldsymbol{x}_k$
  - observation: $\boldsymbol{z}_k$
  - command: $\boldsymbol{u}_k$
  - map: $\boldsymbol{m}$
- models
  - motion: $p(\boldsymbol{x}_{k+1} \mid \boldsymbol{x}_k, \boldsymbol{u}_{k+1})$
  - observation: $p(\boldsymbol{z}_k \mid \boldsymbol{x}_k, \boldsymbol{m})$
- inference
  - prediction:
    $$p(\boldsymbol{x}_{k+1}, \boldsymbol{m} \mid \boldsymbol{z}_{1:k}, \boldsymbol{u}_{1:k+1}) = \sum_{\boldsymbol{x}_k} p(\boldsymbol{x}_{k+1} \mid \boldsymbol{x}_k, \boldsymbol{u}_{k+1}) p(\boldsymbol{x}_k, \boldsymbol{m} \mid \boldsymbol{z}_{1+k}, \boldsymbol{u}_{1:k})$$
  - update:
    $$p(\boldsymbol{x}_{k+1}, \boldsymbol{m} \mid \boldsymbol{z}_{1:k+1}, \boldsymbol{u}_{1:k+1}) \propto p(\boldsymbol{z}_{k+1} \mid \boldsymbol{x}_{k+1}, \boldsymbol{m}) p(\boldsymbol{x}_{k+1}, \boldsymbol{m} \mid \boldsymbol{z}_{1:k}, \boldsymbol{u}_{1:k+1})$$

EKF-SLAM

EKF-SLAM

- landmark map: 2D positions
- Extended Kalman filter on both pose and landmarks
- data association problem:
  - identify which observation corresponds to which map landmark

EKF-SLAM

EKF-SLAM

- landmark map: 2D positions
- Extended Kalman filter on both pose and landmarks
- data association problem:
  - identify which observation corresponds to which map landmark

Inference

- as in a Kalman filter
  - multiplication and inverse of covariance matrices and Jacobians
  - cubic complexity in number of landmarks

Fast-SLAM

Fast-SLAM

- landmark map
- or occupancy grid
- particle filter

Fast-SLAM

Fast-SLAM

- landmark map
- or occupancy grid
- particle filter

Inference

- factorization (Rao-Blackwellization)

$$p(\boldsymbol{x}_{0:k}, \boldsymbol{m} \mid \boldsymbol{z}_{1:k}, \boldsymbol{u}_{1:k}) = p(\boldsymbol{m} \mid \boldsymbol{x}_{0:k}, \boldsymbol{z}_{1:k}) p(\boldsymbol{x}_{0:k} \mid \boldsymbol{z}_{1:k}, \boldsymbol{u}_{1:k})$$

- (simplified) algorithm
  - draw $\boldsymbol{x}_{k+1}^{(i)} \sim \pi(\boldsymbol{x}_{k+1} \mid \boldsymbol{x}_{0:k}^{(i)}, \boldsymbol{z}_{1:k}, \boldsymbol{u}_{1:k+1})$ to augment $\boldsymbol{x}_{0:k}^{(i)}$
  - update weight: $w_{k+1}^{(i)} = w_k^{(i)} \frac{p(\boldsymbol{z}_{k+1} | \boldsymbol{x}_{0:k+1}^{(i)}, \boldsymbol{z}_{1:k}) p(\boldsymbol{x}_{k+1}^{(i)} | \boldsymbol{x}_k^{(i)}, \boldsymbol{u}_{k+1})}{\pi(\boldsymbol{x}_{k+1} | \boldsymbol{x}_{0:k}^{(i)}, \boldsymbol{z}_{1:k})}$
  - resampling
  - update of the map for each particle (with particle pose)

Fast-SLAM

Fast-SLAM

- landmark map
- or occupancy grid
- particle filter

Inference

- factorization (Rao-Blackwellization)

$$p(\boldsymbol{x}_{0:k}, \boldsymbol{m} \mid \boldsymbol{z}_{1:k}, \boldsymbol{u}_{1:k}) = p(\boldsymbol{m} \mid \boldsymbol{x}_{0:k}, \boldsymbol{z}_{1:k}) p(\boldsymbol{x}_{0:k} \mid \boldsymbol{z}_{1:k}, \boldsymbol{u}_{1:k})$$

- (simplified) algorithm
  - draw $\boldsymbol{x}_{k+1}^{(i)} \sim p(\boldsymbol{x}_{k+1} \mid \boldsymbol{x}_k^{(i)}, \boldsymbol{u}_{k+1})$ to augment $\boldsymbol{x}_{0:k}^{(i)}$
  - update weight: $w_{k+1}^{(i)} = w_k^{(i)} p(\boldsymbol{z}_{k+1} \mid \boldsymbol{x}_{0:k+1}^{(i)}, \boldsymbol{z}_{1:k})$
  - resampling
  - update of the map for each particle (with particle pose)

Conclusion on SLAM

Conclusion on SLAM

- integration of localization and mapping
- real-world issue with pose uncertainty and no map
- two approaches
  - asynchronous optimization of pose and map
  - synchronous inference on pose (sequence) and map

03

Conclusion

Conclusion

Mapping

- several kinds of map
- can be easy with known pose
- several uses of the maps

SLAM

- real-world situation
- several approaches for different cases
- inference or optimization to build the map

Bibliography

SLAM

- introduction
  - Durrant-Whyte and Bailey, *Simultaneous Localization and Mapping: Part I*, RAM 2006.
  - Bailey and Durrant-Whyte, *Simultaneous Localization and Mapping: Part II*, RAM 2006.
- particle filter SLAM
  - Dissanayake *et al.*, *A solution to the simultaneous localisation and mapping (SLAM) problem*, IEEE Trans. Robot. Automat. 2001.
  - Montemerlo *et al.*, *Fast-SLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that converges*, IJCAI 2003.
  - Grisetti *et al.*, *Improved techniques for grid mapping with Rao-Blackwellized particle filters*, TRO 2007

Bibliography

Vision

- Klein and Murray, *Parallel tracking and mapping for small AR workspaces*, ISMAR 2007.
- Scaramuzza and Fraundorfer, *Visual odometry*, RAM 2011.
- Fraundorfer and Scaramuzza, *Visual odometry. Part II*, RAM 2012.

Books

- Thrun *et al.*, *Probabilistic Robotics*, MIT Press, 2005.
- Siciliano *et al.*, *Springer Handbook of Robotics*, 2nd ed., Springer, 2016.

Thanks for your attention

Questions?