Robots Autónomos

The Cartographer

Javier de Lope Asiaín
Departamento de Inteligencia Artificial
ETSI Informáticos (UPM)

Overview

The pose graph optimization approach uses lidar scans and attach them to appropriate nodes in an underlying pose graph, which is updated when new scan inputs while the robot explores the world.

The Cartographer method [2] follows this approach and provides a real-time indoor mapping that can generate 2D grid maps with 5 cm resolution.

The scans are inserted into local world areas or *submaps* at the best estimated position. Then, the scans are correlated using a *scan matching*, which happens against recent submaps (recent scans).

Overview (cont.)

An error on the pose estimates is accumulated over time, thus a *pose optimization* process is regularly run to cope with it.

When no new scans are inserted in a submap, it takes part in scan matching for *loop closure*, where scans overlap previously mapped regions, and optimizes the node poses in the pose graph.

The loop closure scan matching happens quicker than the new scans are added (soft real-time constraint). It is achived by using a branch-and-bound approach and several precomputed grids per finished submap.

Submap construction

First, each scan is matched against a small local area of the world or *submap* using a non-linear optimization that aligns the scan with the submap. The process is called *scan matching*.

The scan matching accumulates error over time, which is later removed by the global approach.

A pose $\xi = [x, y, \theta]^T$ is parametrized by a translation (x, y) and an angle θ in a coordinate frame.

Submap construction is the iterative process of repeatedly aligning scan and submap coordinate frames.

Scans

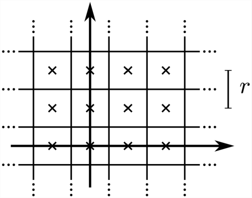
Considering the origin of a scan at $O \in \mathbb{R}^2$ the information about the scan points is $H = \{h_k\}_{k=1,\dots,K}, h_k \in \mathbb{R}^2$.

The pose ξ of the scan frame in the submap frame is represented as the transformation T_ξ defined as

$$T_\xi p = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} p + \begin{pmatrix} x \\ y \end{pmatrix}$$

Submaps

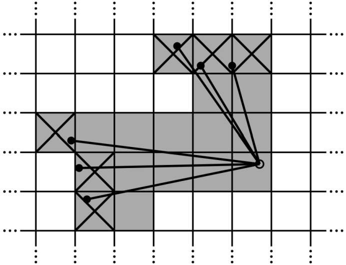
A few consecutive scans are used to build a submap. The submaps take the form of probability grids $M : r\mathbb{Z} \times r\mathbb{Z} \to [p_{min}, p_{max}]$, which map from discrete grid points at a given resolution r to values (probability that a grid point is occupied).



Updating probability grids

Whenever a scan is to be inserted into the probability grid, a set of grid points for *hits* and a disjoint set for *misses* are computed.

- For every hit, the closest grid point into the hit set is inserted.
- For every miss, grid points associated to pixels that intersects one of the rays between the scan origin and each scan point (excluding grid points already in the hit set) are inserted.



Scan matching

The scan poses ξ are optimized relative to the current local submap using Ceres-based scan matcher [1], which is responsible for finding a scan pose that maximizes the probabilities at the scan points in the submap (nonlinear least squares problem):

$$\arg\min_{\xi} \sum_{k=1}^{K} (1 - \mathcal{S}(T\xi h_k))^2$$

where T_ξ transforms h_k from the scan frame to the submap frame according to the scan pose.

$\mathcal{S} : \mathbb{R}^2 \to \mathbb{R}$ is a smooth version of the probability values in the local submap (a bicubic interpolation is used). Mathematical optimization of this smooth function usually gives better precision than the resolution of the grid.

Loops closure

As scans are only matched against a submap containing recent scans, the approach described above slowly accumulates error. The Sparse Pose Adjustment method [3] is used to reduce this error.

Basically, the new scans are compared with the stored ones in submaps to find a good match, which are considered for loop closing. When one is found, the corresponding pose is added to the optimization problem.

Optimization problem

Loop closure optimization also is defined as a nonlinear least squares problem (the solution is also computed with Ceres):

$$\underset{\Xi^m,\Xi^s}{\arg\min}\ \frac{1}{2}\sum_{i,j}\rho\left(E^2(\xi_i^m,\xi_j^s;\Sigma_{ij},\xi_{ij})\right)$$

where the submap poses $\Xi^m = \{\xi_i^m\}_{i=1,\dots,m}$ and the scan poses $\Xi^s = \{\xi_j^s\}_{i=1,\dots,n}$ are optimized given some constraints added by relative poses ξ_{ij} and associated covariance matrices Σ_{ij}.

For a pair of submap i and scan j, the pose ξ_{ij} describes where in the submap coordinate frame the scan was matched, and the residual E for such a constraint is computed.

Moerover, a loss function ρ is used to reduce the influence of outliers that can appear when scan matching adds incorrect constraints to the problem (ie., locally symmetric environments).

Branch-and-bound scan matching

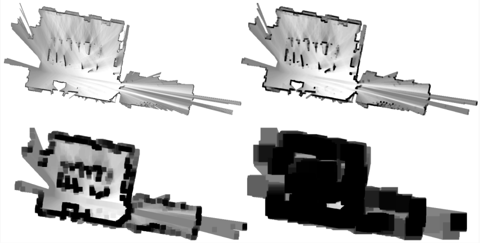
A branch-and-bound approach is used to compute the pixel match over larger search windows.

The main idea is to represent subsets of possibilities as nodes in a tree where the root node represents all possible solutions.

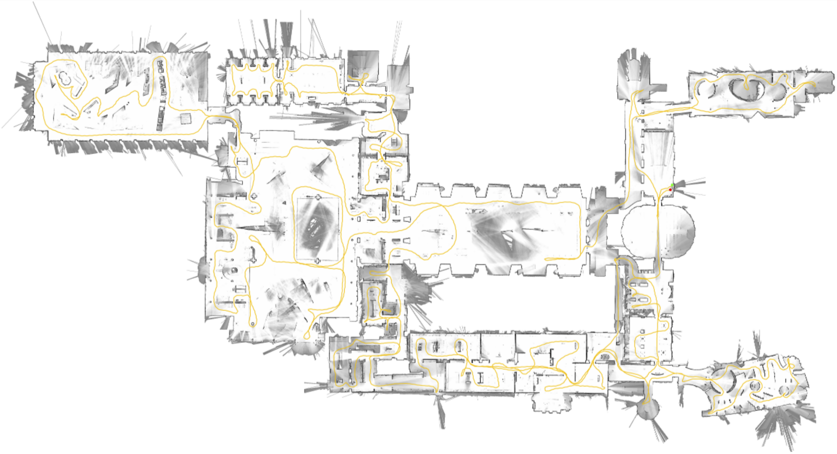
Each node is described as $(c_x, c_y, c_\theta, c_h) \in \mathbb{Z}^4$ (at height c_h exist to $2^{c_h} \times 2^{c_h}$ translations but a specific rotation). Succesor nodes form a partition of their parent and c_h is reduced. Leaf nodes are singletons that represent a single feasible solution ($c_h = 0$).

Branch-and-bound scan matching (cont.)

The algorithm follows a depth-first search and its efficiency depends on how the tree is pruned. To improve it, grids for possible heights are precomputed and used as upper bounds.



Deutsches Museum



References

S. Agarwal, K. Mierle, and Others.
Ceres solver.
`http://ceres-solver.org`.

W. Hess, D. Kohler, H. Rapp, and D. Andor.
Real-time loop closure in 2D LIDAR SLAM.
In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1271–1278, 2016.

K. Konolige, G. Grisetti, R. Kümmerle, W. Burgard, B. Limketkai, and R. Vincent.
Efficient sparse pose adjustment for 2D mapping.
In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 22–29, 2010.