Implementing a SLAM back-end based on least squares error minimization

Supervised by: Cyrill Stachniss

1 Short Description

The goal of this project is to develop a complete SLAM back-end based on the foundations on least squares SLAM presented in the lecture (lessons “Least Squares” and “Graph-based SLAM with Least Squares”). To simplify the developments an Octave framework is provided on the course website.

2 Input

Pose-graph, specified in form of a text file providing the information about nodes and edges. The format is:

```
VERTEX2 id x y theta
EDGE2 vertex1_id vertex2_id delta_x delta_y delta_theta \
      inf_xx inf_xy inf_yy inf_tt inf_xt inf_yt
```

3 Output

A pose-graph in which the configuration of the nodes is the least squares solution.

4 Suggested Work-Plan

- Make sure you have Octave installed.
- Download und unpack the archive with the Octave framework.
- Open the file ls_slam.m.
- Implement the corresponding function (see below).
- The test routine for executing you SLAM code is *LSSlamTest*.
- Test your approach given the three datasets provided as well in the archive.

5 Key Steps for the Implementation

- Define the error function.
- Compute the Jacobian of the error function.
- Setup the linear system.
- Fix one note as the reference frame.
- Transform the $\mathbf{H}$ matrix into a sparse one using the Octave function $sparse())$.
- Solve the linear system using the backslash operator.
- Update the state vector (and normalize the angles to be within $-\pi$ and π).
- Iterate.