Practical Course WS 2010
Simultaneous Localization and Mapping

Cyrill Stachniss

University of Freiburg, Germany
Topics of this Course

- SLAM - simultaneous localization and mapping
  - What does the world look like?
  - Where am I given my world model?

- Task is to build a complete robotic SLAM system that operates on
  - Wheeled base (e.g. Pioneer robot)
  - Laser range scanner (SICK LMS or similar)

- Use of real world data
Goal of this Course

- Hands-on development of a robotic mapping system
- Deeper understanding of the SLAM problem
- Practical programming experience
- Team work
- First experience in planning a software project
Project Structure

- Teams of three people
- Everyone has an own task/component to develop within the project
- Team members are supposed to help each other (tasks may not be equally difficult)
- Components interact via predefined interfaces
Requirements (1)

- Programming skills are essential
- Ability to work in a team
- Knowledge of “Introduction to Mobile Robotics”
- Useful but not essential “Robotics 2”
- Important topics are
  - SLAM, mapping with known poses, laser scanners, scan matching/ICP, error minimization, linear algebra essentials, ...
Requirements (2)

- Main programming language is C++
- Development under Linux (Libraries tested with Ubuntu 10.04)
- Use of versioning tools such as subversion
- Relevant libraries: aisnavigation/aislib, Qt, qglviewer, csparse, cholmod, eigen
- aisnavigation provides: aislib, qglviewer, csparse, eigen
Versioning Tools

- Extremely useful for cooperative development and version tracking
- Stores every change made to the code
- Allows to go back to any intermediate revision
- Can merge different versions
- Inherently multi-user
- Standard tools are subversion or git
- In this practical course, **subversion or git** have to be used
Meetings

- Weekly meetings: Wed 10:15 in bldg 51, HS 03 026
- Each group has to provide a short report presentation (3-7 min) at each meeting
- Each group has to write a brief, informal summary (to be stored in the svn)
- Each group should present the expected progress for the next week
- Each group should evaluate the own progress
Topics of this Course in more Detail

- SLAM - simultaneous localization and mapping
  - What does the world look like?
  - Where am I given my world model?

- Task is to build a complete robotic SLAM system that operates on
  - Wheeled base (e.g. Pioneer robot)
  - Laser range scanner (SICK LMS or similar)
Robot Mapping: “SLAM”

- Ignoring motion and sensor uncertainty leads to inconsistent maps
- Chicken-or-egg problem: Map needed for localization and vice versa
- SLAM = simultaneous localization and mapping
Robot Mapping

- Constraints connect the poses of the robot while it is moving (odometry)
- Constraints are inherently uncertain

Robot pose (x, y, z, yaw, pitch, roll) → Constraint
Robot Mapping

- Observing previously seen areas defines constraints between non-successive poses
- Constraints are inherently uncertain
Idea of Graph-based SLAM

- Use a **graph** to represent the problem
- Every **node** in the graph corresponds to a **pose** of the robot during mapping
- Every **edge** between two nodes corresponds to a **spatial constraints** between them

**Goals:**
- Build the corresponding graph from sensor data
- Find a configuration of the nodes that minimize the error introduced by the constraints
Example

- **Goal:** Find the arrangement of the nodes that satisfies the constraints best

An initial configuration (KUKA production hall 22)
Example

- **Goal:** Find the arrangement of the nodes that satisfies the constraints best

An initial configuration (KUKA production hall 22)
Example

- **Goal:** Find the arrangement of the nodes that satisfies the constraints best

Input

Maximum likelihood configuration
Example

- **Goal:** Find the arrangement of the nodes that satisfies the constraints best

Input

Maximum likelihood configuration
Resulting Tasks

1. SLAM front-end
   - Interpretation of the raw sensor data
   - Scan-matching
   - Finding loop closures

2. SLAM back-end
   - Computing the optimal graph configuration
   - Least squares error-minimimization

3. User interface
   - Visualization of graphs, maps, sensor data
   - Graph editor
   - File import/export
Relevant Papers/Tutorials

- Pose-graph optimization: “Tutorial on Graph-based SLAM” by Grisetti, Kuemmerle, Stachniss, Burgard

- Advanced pose-graph optimization: “Hierarchical Optimization on Manifolds for Online 2D and 3D Mapping” by Grisetti, Kuemmerle, Stachniss, Frese, Hertzberg
  http://www.informatik.uni-freiburg.de/~stachnis/pdf/grisetti10icra.pdf

- Scan matching: ”Real-Time Correlative Scan Matching” by Edwin Olson
  http://april.eecs.umich.edu/pdfs/olson2009icra.pdf

- Advanced loop closing: Chapter 3 of “Robust and Efficient Robotic Mapping” by Olson

- Qt Documentation/Tutorials
  http://doc.qt.nokia.com/4.6/
Relevant Resources

- **aisnavigation** – a not yet publicly available toolbox for navigation-related problems
- **aislib** – basic tools (math, gridmap, graph, posegraph, logfile, ...)
- **Qt/qglviewer** – the probably best framework for user interfaces
- **eigen** – a math toolbox
- **csparse/cholmod** – tools for sparse matrix operations
- There are SLAM libraries available (e.g. see [www.openslam.org](http://www.openslam.org)) but they should not be used...
A Note in Module Interfaces

- Not everyone in this course will continue it up to the end
- Some teams lose an important component
- We will have a global interface for the data exchange between components
- Most simple thing: exchange data only via simple text-files
- File specification will be provided
... That’s It

- New practical project
- Will leads to a state-of-the-art solution in robot mapping
- I hope you like it
- It is quite some work to realize it
Now, Setup Your Team

- Find your team mates
- Decide on the task assignment
- Get familiar with your task (read!)
- Define your own milestones
- Breakdown milestones into tasks
Contact

- Contact us whenever you have problems, questions, or ideas.
- Best is via E-Mail:
  stachnis@informatik.uni-freiburg.de
  kretzsch@informatik.uni-freiburg.de
- Office: Building 79, ground floor
- If you have serious problems, contact us as soon as possible (the other team members depend on you!)