Environment Mapping for Autonomous Driving into Parking Lots

R. Luca, F. Troester, R. Gall; Heilbronn University in Germany
C. Simion; Lucian Blaga University of Sibiu in Romania
Valeo; car components manufacturer in Germany
Presentation Outline

1. Introduction
2. Related Work
3. Hardware/software utilized
4. Algorithms
5. Conclusion
Problem: How to collect information and utilize in a Simultaneous Localization and Mapping (SLAM) algorithm?

• End use: Navigate in unknown environment with the aim of safe self parking on a standard parking lot, through the use of external sensors.

• Main Focus: Reduce data, by algorithms, while maintaining enough map complexity to accomplish task.
Related Work

- Identification Convexity Rule for Shape Decomposition Based on Discrete Contour Evolution.

- Map building for a mobile robot equipped with a 2d laser rangefinder. Integration of numerous local maps into one global map to represent the whole environment observed by the robot during navigation.

- Precise positioning using model-based maps.
  The basis for this paper is using the environment and location of matching previous scans to interpret the position of the mobile robot with respect to fixed landmarks.

  compare 6 popular algorithms for speed, complexity, correctness and precision in robots

- A Method for Building Small-Size Segment-Based Maps
  using a map manager to organize smaller maps into a global map.
Hardware/Software Utilized

SICK LD1000
Laser Measurement System

- Rotating scanner head
  - 5-10 Hz
  - Angular resolution 0.125°
- Supports CAN standard 2.0A
  - Data transmission between 10 bit/s and 1 Mbit/s.
  - Can configure ID for priority on bus
  - Used to transmit distances only
- Digital Outputs
- Emits laser pulses at a max frequency of 14.4 kHz (14,400 per second)
  - Measurement range up to 250m (~820 ft.)
PC/104

- An Embedded Computer Standard
  - Defined form factor and bus.
  - Intended for specialized embedded computing environments dependent on reliable data acquisition.

- Modules stack together like building blocks
  - Typically includes a motherboard, analog-to-digital converter and digital I/O for data acquisition.

- Constraints
  - 3.55 x 3.775 inches
  - Height is typically constrained to the boundaries of the connectors.
Hardware/Software Utilized

xPC Target

- Enables the execution of Matlab/Simulink models on the PC104 system for real-time testing.

- It provides a library of drivers, a real-time kernel and a host target interface for real time monitoring.

- Download code generated by Simulink to the PC104 target via the communications link.
Map Building Procedure

1. 180° scans of the environment as an input to the system.

2. Creates an array of points which is sorted by the laser angle.

3. Array structure

\[ DS1 = [n, x_1, y_1, \ldots, x_n, y_n] \]
### Map Building Procedure

4. Clustering Module
   - Grouping of neighbors
   - New clusters begin when the distance is greater than a predefined parameter.
   - Sorts the clusters.

![Figure 6. Theoretical cluster formation on obstacle edges](image)

**Figure 4. Obtaining the sorted DS1**

---

\[ DS2 = [p_{n1}, x_{on1}, y_{on1}, \ldots , x_{on2}, y_{on2}; \ldots ] \]

---

The obtained sorted data points are further processed in the line extraction algorithm. Figure 5 shows obstacle edges identified within the intersection of the flow of data from sensor until reaching the clustering module.

The reduction of data becomes consistent when clustering occurs, merging points and removing points that are important for the line segment generation, where only the beginning and ending line points are relevant. Scanning the same object from different positions generates new points that are generated between the existent (intermediate) points.

Because of the variable distances between clusters, until reaching the final map structure data reduction becomes effective only when the number of steps is placed on variable number of points.

Considering shape information obtained by a range sensor, mapping becomes effective only when the number of steps is placed on variable number of points.

**Figure 6. Theoretical cluster formation on obstacle edges**

Theoretical cluster formation on obstacle edges is shown in the diagram. The clusters are defined in the algorithm, representing separate clusters.

Separate points representing a higher distance than the one established by the parameter are placed in new lines of a matrix. The sorted clusters are merged together in short line segments, ensuring a reduced processing time and a unified order of the data, which requires the merge and sorting of the clusters.

Therefore, the next module after clustering merges points of the segments, keeping the reducing effect of doubled data, which requires the merge and sorting of the clusters.
Map Building Procedure

5. Lines Module
   - Discrete Contour Evolution, DCE, algorithm
   - First remove doubled data
   - Linear regression is applied
   - Endpoints maintained, intermediate forgotten.

6. DS2 matrix created

\[ DS2 = [n1, x_1, y_1, ..., x_{n1}, y_{n1}; n2, x_1, y_1, ..., x_{n2}, y_{n2}; ...] \]

Figure 7. Reducing DS1 to relevant data DS2

Figure 8. Reducing DS1 to relevant data DS2
### Algorithms

#### Map Building Procedure

7. Map Manager
   - Vehicle Odometry Correction
     uses previous lines to reskew
   - Segmented Maps
     for occlusion and doubled data(segmented)

![Figure 10. Occlusion effect](image)

![Figure 9. DS2 data representation reduced by lines and poly-lines.](image)
Further Use/Research

- Implement an algorithm responsible for object avoidance and for the calculation of the shortest path during parking procedures.
- Remove laser scanner and replace with network of ultrasonic sensor cells.
- Be able to safely autonomously drive the vehicle to a parking lot, based off potential field theory.

For defining the odometry specific to parking procedures, we assume that the single wheel model is reliable to our purpose. The middle point of the rear axis of the vehicle is being tracked and also represents the reference point for the sensor infrastructure design. The driving path is calculated within the incremental sensors installed on each rear wheel axis and the steering wheel angle sensor.

The first task of the autonomous vehicle consists of Mapping. It represents the basic component of the Simultaneous Localization and Mapping (SLAM) concept. This is performed using the SICK LD 1000 laser sensor for environment scanning and incremental map building.

The real-time processing of data has to be considered while the implementation on the PC-104 system. The communication between the laser scanner and the computation unit is made through the CAN I/O port. After reading the data, the computation unit programmed with xPC, the embedded component of Matlab/Simulink, executes algorithmic data reduction for the further Mapping process.

For short range object identification, we use extended ultrasonic cells. Each cell comprises two sensors capable of measuring the closest surrounding area, specific for completing parking procedures. The blind zone of the sensor measuring far is completed by another sensor having a shorter measuring range. The obtained data is also processed by the PC-104 computation unit and used for the parking procedures.

The visualization of data is transferred to a host PC (Laptop) via wireless protocol. We define as target the PC-104 system running the real-time computation. Only relevant data is being kept and processed in the target PC while applying the on-line incremental map building algorithms.

**IV. Map Building Procedure**

We provide a method for map building by considering the acquired scans the input for the system. Each scan consists of 360 points and covers the area in front of the vehicle by indexing a laser beam each 0.5 unit until reaching 180 degrees.

After reaching a number of m scans acquired (in this method, 10 scans=1 cycle), the system creates an array of intersection points which are sorted by angle. The array is structured as following:

$$DS1 = [n, x_1, y_1, \ldots, x_n, y_n]$$

While $n$ represents the number of points generated from the sensor by scanning the environment, the sorted array becomes the input for the clustering module, which operates the points in clusters. Additional sorting of the array of points is necessary for the dynamical map building. As shown in Figure 6.
Conclusion

• Data reduction represents a common solution to mobile robot navigation

• Utilizing real time computation using embedded software data reduction is accomplished.

• Object identification and data reducing algorithms are used and intermediate points/lines are discarded, reducing memory required.
References


