6D SLAM with Kurt3D

Andreas Nüchter, Kai Lingemann, Joachim Hertzberg

University of Osnabrück, Institute of Computer Science
Knowledge Based Systems Research Group
Albrechtstr. 28, D-49069 Osnabrück, Germany
{nuechter|lingemann|hertzberg}@informatik.uni-osnabrueck.de

Abstract. 6D SLAM (Simultaneous Localization and Mapping) or 6D Concurrent Localization and Mapping of mobile robots considers six dimensions for the robot pose, namely, the x, y and z coordinates and the roll, yaw and pitch angles. Robot motion and localization on natural surfaces, e.g., when driving with a mobile robot outdoor, must regard these degrees of freedom. 3D (6 DOF) scan matching, combined with a heuristic for closed loop detection and a global relaxation method, results in a highly precise mapping system for outdoor environments. The mobile robot Kurt3D is capable to run the mapping process with its on-board sensors and computers and is used to digitalize different environments. This paper summarizes our previous research.

1 Introduction

Automatic environment sensing and modeling is a fundamental scientific issue in robotics, since the presence of maps is essential for many robot tasks. Manual mapping of environments is a hard and tedious job: Thrun et al. report a time of about one week hard work for creating a map of the museum in Bonn for the robot RHINO [9]. Especially mobile systems with 3D laser scanners that automatically perform multiple steps such as scanning, gaging and autonomous driving have the potential to greatly improve mapping. Many application areas benefit from 3D maps, e.g., industrial automation, architecture, agriculture, the construction or maintenance of tunnels and mines and rescue robotic systems.

The robotic mapping problem is that of acquiring a spatial model of a robot's environment. If the robot poses were known, the local sensor inputs of the robot, i.e., local maps, could be registered into a common coordinate system to create a map. Unfortunately, any mobile robot's self localization suffers from imprecision and therefore the structure of the local maps, e.g., of single scans, needs to be used to create a precise global map. Finally, robot poses in natural outdoor environments necessarily involve yaw, pitch, roll angles and elevation, turning pose estimation as well as scan registration into a problem with six mathematical dimensions.

2 State of the Art

State of the art for metric maps are probabilistic methods, where the robot has probabilistic motion models and uncertain perception models. Through integration of these two distributions with a Bayes filter, e.g., Kalman or particle filter, it is possible to localize the robot. Mapping is often an extension to this estimation problem. Beside the robot pose, positions of landmarks are estimated. Closed loops, i.e., a second encounter of a previously visited area of the environment, play a special role here: Once detected, they enable the algorithms to bound the error by deforming the mapped area to yield a topologically consistent model. However, there is no guarantee for a correct model. Several strategies exist for solving SLAM. Thrun surveys existing techniques, i.e., maximum likelihood estimation, expectation maximization, extended Kalman filter or (sparsely extended) information filter SLAM [10].

SLAM in well-defined, planar indoor environments is considered solved, but 6D SLAM still proposes a challenge, since several strategies become infeasible, e.g., with 6 degrees of freedom the matrices in Kalman or information filter SLAM grow more rapidly and a multi hypothesis approach would require too many particles. Therefore, 3D mapping systems [2–4, 6, 7] often rely on scan matching approaches.

3 Kurt3D

3.1 The 3D laser range finder.

The 3D laser range finder (Fig. 1) [7] is built on the basis of a SICK 2D range finder by extension with a mount and a small servomotor. The 2D laser range finder is attached in the center of rotation to the mount for achieving a controlled pitch motion with a standard servo.

The area of up to $180°(\mathrm{h}) \times 120°(\mathrm{v})$ is scanned with different horizontal (181, 361, 721) and vertical (128, 256, 400, 500) resolutions. A plane with 181 data points is scanned in 13 ms by the 2D laser range finder (rotating mirror device). Planes with more data points, e.g., 361, 721, duplicate or quadruplicate this time. Thus a scan with 181 $\times$ 256 data points needs 3.4 seconds. Scanning the environment with a mobile robot is done in a stop-scan-go fashion.

3.2 The mobile robot.

Kurt3D (Fig. 1) is a mobile robot with a size of 45 cm (length) $\times$ 33 cm (width) $\times$ 29 cm (height) and a weight of 22.6 kg. Two 90 W motors are used to power the 6 skid-steered wheels, whereas the front and rear wheels have no tread pattern to enhance rotating. The core of the robot is a Pentium-Centrino-1400 with 768 MB RAM and Linux.

Fig. 1: Kurt3D.

4 6D SLAM

To create a correct and consistent environment map, 3D scans have to be merged into one coordinate system. This process is called registration. If the robot carrying the 3D scanner were localized precisely, the registration could be done directly based on the robot pose. However, due to the imprecise robot sensors, self localization is erroneous, so the geometric structure of overlapping 3D scans has to be considered for registration. As a by-product, successful registration of 3D scans relocalizes the robot in 6D, by providing the transformation to be applied to the robot pose estimation at the recent scan point.

Kurt3D's SLAM algorithm consists of four steps, that are explained in the following subsections.

4.1 Odometry extrapolation

Thh odometry is extrapolated to 6 degrees of freedom using previous registration matrices, i.e., the change of the robot pose $\Delta\mathbf{P}$ given the odometry information $(x_n, z_n, \theta_{y,n})$, $(x_{n+1}, z_{n+1}, \theta_{y,n+1})$ and the registration matrix $\mathbf{R}(\theta_{x,n}, \theta_{y,n}, \theta_{z,n})$ is calculated by solving:

$$\begin{pmatrix} x_{n+1} \\ 0 \\ z_{n+1} \\ 0 \\ \theta_{y,n+1} \\ 0 \end{pmatrix} = \begin{pmatrix} x_n \\ 0 \\ z_n \\ 0 \\ \theta_{y,n} \\ 0 \end{pmatrix} + \left(\begin{array}{c|c} \mathbf{R}(\theta_{x,n}, \theta_{y,n}, \theta_{z,n}) & \mathbf{0} \\ \hline \mathbf{0} & \begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix} \end{array} \right) \cdot \underbrace{\begin{pmatrix} \Delta x_{n+1} \\ \Delta y_{n+1} \\ \Delta z_{n+1} \\ \Delta\theta_{x,n+1} \\ \Delta\theta_{y,n+1} \\ \Delta\theta_{z,n+1} \end{pmatrix}}_{\Delta\mathbf{P}}.$$

4.2 Calculating Heuristic Initial Estimations for ICP Scan Matching

For the given two sets M and D of 3D scan points stemming from the 3D scans, our heuristic computes two octrees based on these point clouds. The octrees rigid transformations are applied to the second octree, until the number of overlapping cubes has reached its maximum. The transformations are computed in nested loops. However, the computational complexity is reduced due to the fact that we limit the search space relative to the octree cube size. Details can be found in [4].

4.3 Scan Registration

We use the well-known Iterative Closest Points (ICP) algorithm to calculate a rough approximation of the transformation while the robot is acquiring the 3D scans [1]. The ICP algorithm calculates iteratively the point correspondence. In

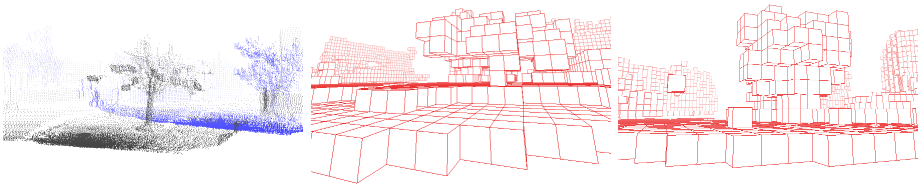

Fig. 2. Left: Two 3D point clouds. Middle: Octree corresponding to the black point cloud. Right: Octree based on the blue points.

each iteration step, the algorithm selects the closest points as correspondences and calculates the transformation $(\mathbf{R}, \mathbf{t})$ for minimizing the equation

$$E(\mathbf{R}, \mathbf{t}) = \sum_{i=1}^{N_m} \sum_{j=1}^{N_d} w_{i,j} \left|\left|\mathbf{m}_i - (\mathbf{R}\mathbf{d}_j + \mathbf{t})\right|\right|^2 , \quad (1)$$

where N_m and N_d, are the number of points in the model set M or data set D, respectively and w_{ji} are the weights for a point match. The weights are assigned as follows: $w_{ji} = 1$, if $\mathbf{m}_i$ is the closest point to $\mathbf{d}_j$ within a close limit, $w_{ji} = 0$ otherwise. The assumption is that in the last iteration step the point correspondences, thus the vector of point pairs, are correct.

4.4 Loop Closing

After matching multiple 3D scans, errors have accumulated and loops would normally not be closed. Our algorithm automatically detects a to-be-closed loop by registering the last acquired 3D scan with earlier acquired scans. Hereby we first create a hypothesis based on the maximum laser range and on the robot pose, so that the algorithm does not need to process all previous scans. Then we use the octree based method presented in section 4.2 to revise the hypothesis. Finally, if a registration is possible, the computed error, i.e., the transformation $(\mathbf{R}, \mathbf{t})$ is distributed over all 3D scans.

4.5 Model Refinement

Based on the idea of Pulli we designed the relaxation method *simultaneous matching* [7]. The first scan is the masterscan and determines the coordinate system. It is fixed. The following three steps register all scans and minimize the global error, after a queue is initialized with the first scan of the closed loop:

1. Pop the first 3D scan from the queue as the current one.
2. If the current scan is not the master scan, then a set of neighbors (set of all scans that overlap with the current scan) is calculated. This set of neighbors forms one point set M. The current scan forms the data point set D and is aligned with the ICP algorithms. One scan overlaps with another iff more than p corresponding point pairs exist. In our implementation, $p = 250$.
3. If the current scan changes its location by applying the transformation (translation or rotation) in step 2, then each single scan of the set of neighbors that is not in the queue is added to the end of the queue. If the queue is empty, terminate; else continue at step 1.

In contrast to Pulli's approach, our method is totally automatic and no interactive pairwise alignment has to be done. Furthermore the point pairs are not fixed [5]. The accumulated alignment error is spread over the whole set of acquired 3D scans. This diffuses the alignment error equally over the set of 3D scans [8].

5 Results and Conclusions

The proposed methods have been tested on various data sets, including test runs at RoboCup Rescue and ELROB. Fig. 3 show two closed loops. 3D animations of the scenes can be found at `http://kos.informatik.uni-osnabrueck.de/download/6Dpre/` and `http://kos.informatik.uni-osnabrueck.de/download/6Doutdoor/`.The loop in the left part of Fig. 3 was closed manually, whereas the right loop was detached automatically.

These large loops require an reliable robot control architecture for driving the robot and efficient 3D data handling and storage methods. In future work we will tackle the emerging topic of map management.

References

1. P. Besl and N. McKay. A method for Registration of 3–D Shapes. *IEEE Transactions on PAMI*, 14(2):239 – 256, February 1992.
2. A. Georgiev and P. K. Allen. Localization Methods for a Mobile Robot in Urban Environments. *IEEE Trans. Robotics and Automation (TRO)*, 20(5), 2004.
3. Martin Magnusson and Tom Duckett. A comparison of 3d registration algorithms for autonomous underground mining vehicles. In *Proc. ECMR*, 2005.
4. A. Nüchter, K. Lingemann, J. Hertzberg, and H. Surmann. Heuristic-Based Laser Scan Matching for Outdoor 6D SLAM. In *KI, Springer LNAI vol. 3698*, 2005.
5. K. Pulli. Multiview Registration for Large Data Sets. In *Proc. 3DIM*, Oct. 1999.
6. V. Sequeira, K. Ng, E. Wolfart, J. Goncalves, and D. Hogg. Automated 3D reconstruction of interiors with multiple scan–views. In *Proc.of SPIE, Electronic Imaging '99, 11th Annual Symposium*, San Jose, CA, USA, January 1999.
7. H. Surmann, A. Nüchter, and J. Hertzberg. An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor en vironments. *Journal Robotics and Autonomous Systems*, 45(3 – 4):181 – 198, December 2003.
8. H. Surmann, A. Nüchter, K. Lingemann, and J. Hertzberg. 6D SLAM A Preliminary Report on Closing the Loop in Six Dimensions. In *Proc. of the 5th IFAC Symp. on Intelligent Autonomous Vehicles (IAV '04)*, Lisabon, Portugal, July 2004.
9. S. Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71, 1998.
10. S. Thrun. Robotic mapping: A survey. In G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann, 2002.

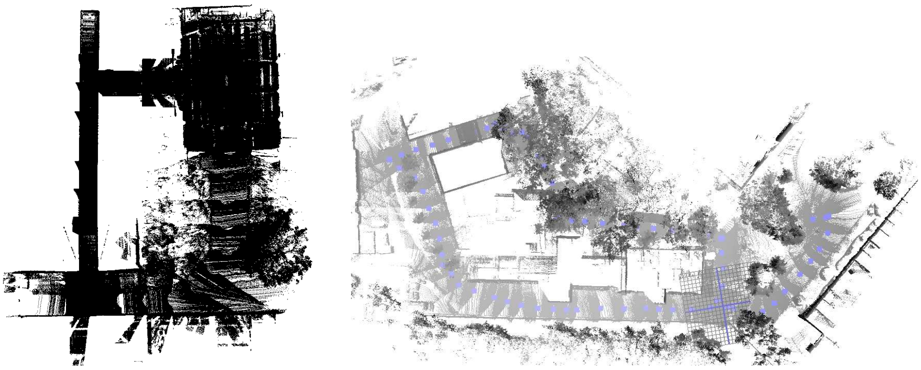

Fig. 3. Two 3D point clouds in top view. Left: Closed loop with 9 million 3D data points. Loop with 7 million points and a path length of over 250 m.