3D Reconstruction for Exploration of Indoor Environments

Jens Wettach and Karsten Berns

Robotics Research Lab, University of Kaiserslautern, Germany
{wettach,berns}@informatik.uni-kl.de

Abstract. Autonomous exploration of arbitrary indoor environments with a mobile robot depends on a reliable self-localization strategy. Existing approaches that use only 2D distance information from e.g. planar laser scanners may fail in highly cluttered areas due to the lack of stable landmark detection. This paper presents an approach for extracting room and furniture primitives from a 3D point cloud by matching shape primitives to the data samples. These basic building blocks can serve as landmarks for relocalization and give hints for interesting places during environmental exploration. Input data is acquired by a tiltable 2D laser scanner in reality and a realistic virtual sensor simulation. In the paper the complete process from sensor data acquisition, data filtering, RANSAC[1] based plane extraction and smoothing is described and tested in simulation and reality.

1 Introduction

Application scenario is an autonomous exploration of indoor environments with mobile robot MARVIN(see figure 1(b)). Its main sensors are a SICK S3000 laser scanner at its front side and a LMS200 scanner at its rear side which provide distances to nearest obstacles within a horizontal measurement plane about 10 cm above ground. Based on these sensors the robot is able to create a topological map of rooms and doors autonomously (see [4]). As this exploration relies on detection of room walls the strategy fails in cluttered areas where the two scanners do not "see" enough wall segments. Thus extraction of room primitives with an additional 3D scanner has come into focus to solve these problems.

The goal here is to create a sensor-based "robot view of the world" by approximating a 3D scene represented as point cloud of distance values with simple primitives as planes, cubes and cylinders (see [5] for a similar approach). These basis building blocks will help the robot to "understand" its environment sufficiently for localization and navigation tasks. Of course, for high-level tasks as fetching objects from a certain desk a matching of human understanding of things and these basic blocks has to be ascertained (long-term goal). By now only plane patches are extracted from the scene as this seems to be sufficient for getting a rough overview of a room and thus will serve as first step for solving the localization problem. The presented approach of plane extraction from an

[1] Random Sample Consensus (see[1])

arbitrary 3D point cloud is mainly based on the Grid-based Segmentation (GBS) algorithm presented in [2] and [3].

The remainder of this paper is organized as follows: the sensor data acquisition is decribed in section 2 and the feature extraction strategy in 3. Experimental results are given in 4, summary and outlook in 5.

2 Data Acquisition

Input for the feature extraction strategy is an arbitrary 3D point cloud as result of distance measurements which are either collected by a rotating 2D laser scanner or by a simulated sensor realized within the SimVis3D framework.

2.1 3D Scanner Device

The 3D scanner consists of a SICK LMS 200 laser range finder mounted on a tilt unit (see figure 1(b)) which enables an external rotation of the whole device around a horizontal axis from -45° to 45°. The scanner measures distances to nearest objects in sight using a pulsed laser beam that is deflected by a rotating mirror within the scanner case. The maximal scan area is 180° with a native angular resolution of 1°. Two consecutive scans are combined with an offset of 0.5° to get a resolution of 0.5°. Thus a complete scan consists of 361 distance values sent to the PC within one scan telegram (see figure 1(a)).

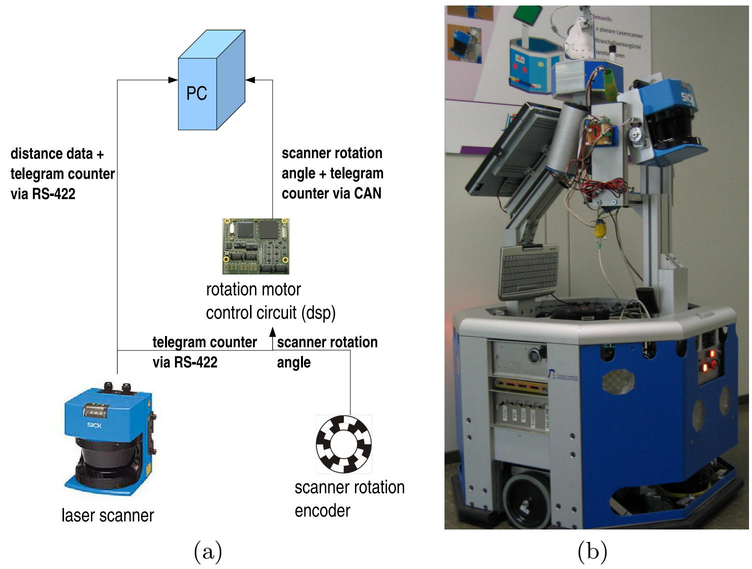


Fig. 1. (a) Fusion of distance data from laser scanner and external scanner tilt angle from DSP, (b) mobile robot MARVIN with mounted 3D scanner device

Each scan telegram is tagged with a counter value (modulo 256). The external rotation of the complete scanner device is controlled by a multi-purpose dsp control circuit (developed at RRLAB) which executes a speed and position control loop. This rotation is measured via an incremental encoder attached to the tilt axis with a resolution of 0.09°. For collecting a 3D point cloud the scanner is rotated continuously from start to end position. During this rotation the sensor collects distance data and sends scan telegrams to the PC via RS-422 link also in a continuous mode.

As the external sensor motion and the internal mirror rotation are independent of each other the problem is how to fuse distance and angular information consistently. As solution the motor control dsp also reads the scan telegrams from the RS-422 link. Whenever it has parsed a complete telegram it extracts the telegram counter and tags the corresponding scanner tilt angle with it. Here it takes into account that the device first collects data from two mirror rotations completely and afterwards sends the telegram. Thus the scanner tilt angle at the begin of each measurement is stored by the dsp program until the corresponding telegram counter has been retrieved and is then sent to the PC via CAN bus.

The data fusion software run in the PC receives scan telegrams from serial input and scanner tilt angles from CAN bus and brings them into correspondence via the telegram counter, using a local history queue. Furthermore, assuming a continuous scanner rotation, the tilt angles at the start and end of each measurement can be propagated to each distance value within one telegram. At this point it has to be taken into account that each measurement is composed of two scanner mirror rotations, that the distance values are retrieved during the first half of each rotation (when the laser beam leaves the device at its front side) and that the full scan angles belong to the first rotation and the half angles to the second.

This fusion strategy allows a complete independent internal (mirror) and external (scanner) rotation as long as the latter one is continuous. For that purpose the data evaluation is stopped at each turning point of the tilt motion which is signaled by the dsp via CAN bus. A dataset of distance values usually comprises measurements of one scan sweep (from start to end tilt angle or vice versa). As each distance value is affected by noise neighboring samples are smoothed locally within each scan using a windowing average filter.

2.2 Virtual 3D Scanner within the SimVis3D Framework

For testing the feature extraction algorithm offline a virtual 3D scanner has been realized within the SimVis3D framework (see [6] for further information). As in reality the 3D sensor uses a tiltable (virtual) 2D scanner providing distance information from its mounting pose in the 3D scene to nearest objects in sight. It is realized as an additional sensor device like the existing camera sensor.

During implementation of the virtual scanner it has become obvious that a straight forward distance calculation based on ray tracing operations is too slow (about 2 s for 361 data points/180° range). Thus the simulation uses a virtual perspective scene camera whose image is rendered to an offscreen buffer. The depth values of this buffer are then evaluated taking the perspective distortion of the camera and the regular scan angle resolution of $0.5°$ into account. Naturally with one camera setting a viewing angle of $180°$ is not possible, so the camera image is rendered from two different positions (for scan angles $0° \dots 90°$ and for $90.5° \dots 180°$). Hereby the camera viewport is adjusted so that its depth buffer resolution matches the distance value requirements optimally. This way also scanners with higher angular range (Hokuyo URG-04LX, SICK S300) can be simulated.

The camera-based distance value calculation strategy is fast enough to simulate a scanner at a rate of about 20 Hz. To make the measurements even more realistic a Gaussian noise is added to each data point. Thus the simulation allows to collect a 3D point cloud of distance measurements as the real device. Figure 3(a) shows a measurement within the virtual RRLAB.

3 Plane Extraction

The main goal of the plane extraction algorithm is to get reliable landmarks as floor, ceiling, walls, doors and big furniture objects like desks and cupboards from the collected 3D point cloud. These features are used for reliable robot localization in highly cluttered areas where the 2D scanners do not provide stable features.

As stated in section 1 the plane extraction procedure is realized according to the strategy presented in [3]. Goal is to approximate the input data by a set of plane patches such that each set of points is represented in a least square sense by its patch and that features represented by a huge amount of samples (e.g. corridor walls) are represented by one big patch. Of course using only planes as shape for 3D environmental approximation is invalid for many objects (e.g. dustbins would be better represented by a cylinder), but is suitable for finding the mentioned room primitives.

In the following a summary of the different steps of the extraction method is given. The complete strategy is described detailed in [7]. Algorithm 1 sketches the whole procedure.

The RANSAC algorithm in step 1 repeatedly calculates planes approximating the local set of points within one cell. In each iteration it randomly selects three (not aligned) points within the local set, calculates a plane through these points and computes the sum of the distances of all other points within the cell to this plane. This sum of distances is the minimization criterion: the calculated plane is used as best plane when its sum of distances is smaller than the one of the best plane calculated during the previous iterations. This procedure is motivated by the fact that the repeated random selection of points and plane approximation converges to a "good" fitting plane if such a plane exists at all (model valid) and the number of iterations is high enough.

Within this work not more than 100 RANSAC iterations are executed to save computation time. If the samples cannot be approximated well by a plane it will not be merged during region growing and thus filtered out in step 5.

Algorithm 1 RANSAC algorithm for plane extraction

Given: a set of 3D distance measurement samples (data points)
Return: a set of planes approximating disjunctive subsets of the input points
Step 1: split the whole 3D scene into a regular grid of cubic cells
Step 2: assign the input points to the corresponding cells
Step 3: calculate fitting plane for each cell:
for every cell **do**
  find approximating plane using RANSAC algorithm (output: best fitting plane after certain number of RANSAC iterations)
  remove outliers with respect to the best RANSAC plane
  calculate least-square fitting plane for inliers
end for
Step 4: region growing - fuse matching planes of neighboring cells:
for every cell **do**
  compare plane parameters with those of all neighboring cells
  if angle between plane normals below angular threshold and distance of center of gravity of points of neighboring cell to plane below distance threshold **then**
    mark both cells as belonging to same region
  end if
end for
for all regions **do**
  calculate best fitting plane for all points of cells belonging to the same region
end for
Step 5: remove small planes supported only by a small amount of cells

The outlier calculation regarding the best RANSAC plane uses an adjustable distance threshold. The plane fitting for all inliers within one cell as well as for all samples of one merged region is calculated via *principal component analysis*. A plane is described by its normal $\overrightarrow{n}$ and distance from origin d (eq. 1).

$$< \overrightarrow{n}, \overrightarrow{x} >= d \tag{1}$$

According to [3] the normal of the least-square fitting plane is the eigenvector of the smallest eigenvalue of covariance matrix A (eq. 2).

$$A = \begin{pmatrix} \sum_{i=0}^{N} w_i x_i^2 & \sum_{i=0}^{N} w_i x_i y_i & \sum_{i=0}^{N} w_i x_i z_i \\ \sum_{i=0}^{N} w_i x_i y_i & \sum_{i=0}^{N} w_i y_i^2 & \sum_{i=0}^{N} w_i y_i z_i \\ \sum_{i=0}^{N} w_i x_i z_i & \sum_{i=0}^{N} w_i y_i z_i & \sum_{i=0}^{N} w_i z_i^2 \end{pmatrix} \tag{2}$$

Here $x_i = x_i^{raw} - \overline{x}$, $y_i = y_i^{raw} - \overline{y}$ and $z_i = z_i^{raw} - \overline{z}$ are input points centered around mean and weights w_i represent measurement uncertainty of the samples (set to 1 as uncertainty is not yet taken into account). d is given by eq. 3

$$d =< \overrightarrow{cog}, \overrightarrow{n} > \tag{3}$$

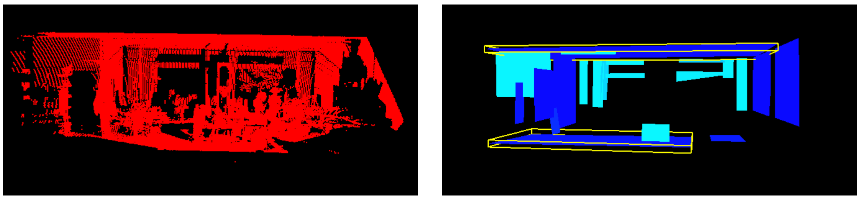
where $\overrightarrow{cog} = (\overline{x}, \overline{y}, \overline{z})^T$ is the center of gravity of points belonging to the fitted plane. As A is a square symmetric matrix its eigenvalues can be computed efficiently via *singular value decomposition*. Table 1 lists all adjustable parameters used by the described algorithm.

step	parameter	value
1	cell size	200 mm
3	min. amount of points in a cell to start RANSAC algorithm	10
3	number of RANSAC iterations	50
3	max. point-to-plane distance for inliers	50 mm
4	angular threshold for normals of neighboring planes	15°
4	distance threshold for cog of one plane to neighboring plane	50 mm
5	min. amount of supporting cells for plane filtering	10

Table 1. parameters used for plane extraction

4 Experiments

The plane extraction algorithm has been tested successfully with real and simulated scanner data. Figure 2(a) shows a 3D scan of a typical scene at RRLAB and figure 2(b) shows the extracted planes. In figures 3(a), 3(b) a similar scenario is shown as simulation.

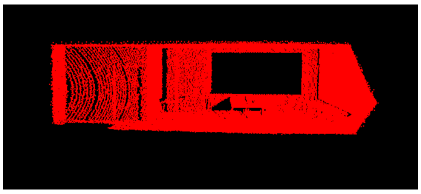


(a) (b)

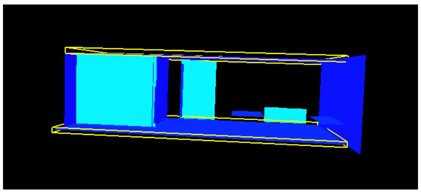
Fig. 2. (a) Real 3D scan of RRLAB indoor scene, (b) extracted planes

The angular range of the tilt motion is -40° ... 40° with a resolution of about 0.3° per scan. The total number of samples is ∼92000 in each case and the time for plane extraction is less than 2 s. In both figures floor and ceiling plane have been marked by their bounding boxes to get a better visual impression. As the minimum number of supporting cells is 10 (ref. table 1), only big planes representing room primitives as walls or large window frames (upper middle part in fig. 2(b)) have been extracted. The (a priori infinite) planes have been clipped by the bounding box of the supporting 3D points. Alternatively the planes can be clipped by the border lines of the supporting cells, but this leads to collections of small faces for diagonal planes which do not intersect a contiguous set of cells.

The simulation experiment within the virtual indoor scene yields similar results as in reality. As smaller objects like chairs, screens and dustbins are not yet included, the extracted planes are more regular and also table tops (middle part of fig. 3(b)) are extracted.



(a)



(b)

Fig. 3. (a) Simulated 3D scan of virtual RRLAB scene, (b) extracted planes

5 Conclusion and Outlook

In this paper a strategy for extracting room primitives from 3D distance measurements has been presented. The applied sensor hardware and a corresponding simulation framework have been introduced and the RANSAC based feature extraction algorithm has been explained. The described test results give an impression how floor, walls and ceiling features are detected even in highly cluttered real-world scenarios. Thus these results can be used as reliable landmarks for robot localization during mapping and for navigation purposes.

Next steps comprise the extraction of additional shapes like cylinders and cuboids (composition of rectangular planes) to model the scene as collection of primitive building blocks. Besides the fusion of features from several consecutive 3D scans (i. e. matching of features from different viewpoints) has to be solved. As long-term goal also some semantic information has to be attached to the detected objects, e. g. a table as cuboid with horizontal top where often objects of daily use are placed. For then the extracted primitives can be matched with human scene understanding and thus help for managing higher level tasks as searching and fetching interesting objects.

References

1. M. A. Fischler and R. C. Bolles. *Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography.* Commun. ACM 24, 1981.
2. J. Weingarten. *Feature-Based 3D SLAM.* PhD thesis, EPFL, September 2006.
3. J. Weingarten, G. Gruener, and R. Siegwart. A fast and robust 3d feature extraction algorithm for structured environment reconstruction. In *Proc. of ICAR 2003.*
4. D. Schmidt, T. Luksch, J. Wettach, and K. Berns. Autonomous behavior-based exploration of office environments. In *Proc. of ICINCO 2006.*
5. R. Schnabel, R. Wahl, and R. Klein. Shape detection in point clouds. Technical report, Computer Graphics Technical Report, 2006.
6. T. Braun, J. Wettach, and K. Berns. A customizable, multi-host simulation and visualization framework for robot applications. In *Proc. of ICAR 2007.*
7. X. Astigarraga. 3d reconstruction of structured indoor environments. Master's thesis, University of Kaiserslautern, March 2007.