3-D-Environment Reconstruction for Mobile Robots using fastSLAM and Feature Extraction

Georg Arbeiter, Jan Fischer, Alexander Verl
Fraunhofer IPA, Nobelstr. 12, 70597, Stuttgart, Germany

Abstract

This paper proposes an algorithm that can be used to reconstruct a 3-D environment on a mobile robot. As sensors, color and time-of-flight cameras are used. 2-D features are extracted from color images and assigned 3-D coordinates. Those are the input for a modified fastSLAM algorithm that is capable of rendering environment maps for small environments online in order to execute manipulation tasks. The method is evaluated on the service robot Care-O-bot® 3.

1 Introduction

3-D environment perception is one of the key technologies for human-scale service robots. A three-dimensional view of the surroundings of a robot is crucial for accomplishing tasks like navigation and manipulation in a fully autonomous way in incompletely known environments. Also, for tele-operation of robots a visualization of the environment in a human-readable way is important for an intuitive user interface. Because the data of state-of-the-art sensors is bound to error and the robot position is usually completely unknown or only roughly known, a transformation of the single sensor readings w.r.t. the robot position does not lead to sufficient results. Therefore, methods that minimize those errors are of great importance. In the past decades, there has been a lot of research in the field of simultaneous localization and mapping (SLAM). Most of the algorithms like EKF SLAM are based on the Bayesian filter theory. A promising approach is fastSLAM [5] that decomposes the SLAM problem into a robot localization problem and a collection of landmark estimation problems. Because of that, fastSLAM has logarithmic complexity with respect to the number of landmarks N whereas EKF SLAM has $O(N^2)$.
Other approaches for 3-D environment modeling use the iterative closest point algorithm (ICP). The ICP tries to find a transformation between two point clouds that minimizes the distance between the points. The algorithm is iterated as long as the distance error is below a given threshold. Nüchter et. al. [2] evaluated many variants of the ICP and were able to develop a method to create 3-D point maps from laser scanner readings while moving the robot with full 6 degrees-of-freedom.
Most of the above mentioned methods were designed for the use of 2-D or 3-D laser range finders. Also, almost every SLAM algorithm is intended to be used mainly for navigation in large-scale environments. As we focus on manipulation tasks in small and dynamic environments, state-of-the-art algorithms cannot be used out-of-the-box. Also, we use different sensors on our robot, the Care-O-bot® 3 [4]. Instead of 3-D laser ranger finders, as used for example on the PR2 [3], the Care-O-bot® 3 is equipped with an agile sensor head. As shown in figure 1, two color and one 3-D time-of-flight (TOF) camera are mounted on the head. Those are different pre-conditions for 3-D environment perception than in most of the other research activities. Because the Care-O-bot® 3 is also equipped with three 2-D laser ranger finders for navigation, it can use those to get a rather good position estimate of the robot. Also, those objects in the environment that can hardly be moved (e.g. walls, kitchen shelves, heavy furniture) are modeled in a static map. This map is used for localization. For manipulation tasks the 2-D map is augmented by 3-D objects obtained from the 3-D perception algorithm. The goal is to create a dense dynamic 3-D map that renders the surroundings of the manipulation area, e.g obstacles that may be in the planned manipulator path or influence possible grasp strategies.

Figure 1: Sensor head of Care-O-bot® 3. Two color cameras and one time-of-flight camera.

The innovation of this paper is the application of the fastSLAM 2.0 [10] algorithm for manipulation tasks on Care-O-bot® 3. Therefore, 2-D features have to be extracted from color camera images and assigned 3-D coordinates with help of the TOF camera. Furthermore, fastSLAM is modified in order to use descriptor-based data association and handle multiple observations per timestep. In that way, a 3-D feature map of the environment is obtained. If the density of the map is not sufficient, the raw data point clouds can be transformed according to the corrected robot position in order to get a consistent point cloud of the whole scene.
The remainder of this paper is organized as follows. Section 2 gives an overview of sensor data preprocessing, followed by Section 3 which describes the feature extraction. Section 4 proposes a modified fastSLAM algorithm for environment reconstruction and Section 5 presents the actual implementation on Care-O-bot® 3. Finally a conclusion and outlook is given in Section 6.

2 Sensor Data Preprocessing

As time-of-flight cameras are inaccurate compared to laser range finders, methods to remove noise and erroneous measurements have to be considered. The main error sources are presented and simulated in [5]. The key effects are flying pixels, wiggling, motion blur and noise. Flying pixels mainly occur on edges and object boundaries due to depth inhomogeneities, wiggling is the systematic deviation error. The influence of those effects on image quality is small, so they are ignored. Motion blur arises from fast camera motions due to temporal integration of the phase image. This problem can be solved by performing slow robot motions during measurements or incremental movements with sensor data acquisition in between.
Noise mainly comes from reflections in the surroundings of the sensor. Other artifacts similar to noise are generated when objects in the sensor's view are further than the non-ambiguity range. Those pixels appear to be much nearer than they are and tend to be scattered. Both noise and out-of-range pixels are characterized by low amplitude and intensity values. Because of that, intensity-based filtering can be performed. The coordinates of a point i are masked with 0 if $I_i < I_{th}$ with the intensity of the point I_i and the intensity threshold I_{th}. As those values can never occur during normal sensor operation, they can be clearly identified in later processing steps.

3 Feature Extraction

Sensor readings for feature extraction come from a color camera, mounted on the sensor head of the mobile robot. A robust feature detector and descriptor for color images is SURF (Speeded Up Robust Features) [6]. It uses a fast-hessian detector to identify keypoints in an image and the SURF descriptor to describe those points. It outperforms the formerly introduced SIFT [7] both in robustness and speed.
As the feature extraction only provides 2-D features in image coordinates, the corresponding 3-D coordinates have to be associated. The use of a colored point cloud [8] makes it possible to assign 3D coordinates to the feature points. Either a stereo camera system or the time-of-flight camera in combination with one color camera can be used to obtain a colored point cloud.

4 FastSLAM for Environment Reconstruction

This section describes the fastSLAM algorithm used for environment reconstruction. FastSLAM 2.0 was first introduced in [10]. It is a particle filter approach to simultaneous localization and mapping (SLAM) which is based on the assumption that if robot's true path were known, the SLAM problem would be an estimation problem of independent landmarks. So it can reduce the complexity in comparison to EKF SLAM to $O(M \log N)$, with M being the number of particles and N the number of landmarks in the map. Because of the particle structure it is able to handle multi-hypothesis data association.
As visual SURF features have a descriptor that consists of 64 values, the original fastSLAM has to be modified with respect to data association. Otherwise the computational cost of the data association would slow down the algorithm significantly. Furthermore, handling of multiple observation per time step is necessary because of the sensors used. Related to [11], all of the detected features are stored in a kd-tree whereas the SURF descriptor is used as search pattern. The leafs of the tree contain a unique identifier for the feature that is assigned during creation of the tree. For associating a feature point, a nearest neighbor search is performed within the tree that returns the ID of the feature with the most similar descriptor.
Because there is no convenient way to add or remove features from the kd-tree, it has to be rebuilt after each association step. That is the reason why global data association is used rather than per-particle association. Doing this in every particle would be inefficient as building a balanced tree out of n points. This has at least $O(n \log n)$ complexity, depending on the search method. Also, as SURF features are robust and yield few misassociations, we don't loose much by neglecting the per-particle association.
Each ID of the associated features is passed to the filter. Within each particle the features are stored in a binary tree and a search is done by ID. In order to detect misassociations, the importance p of the measurement is calculated by

$$p = |2\pi Z|^{-\frac{1}{2}} \exp\{-\frac{1}{2}(z-\hat{z})^T Z^{-1}(z-\hat{z})\}$$

with the actual measurement z, the predicted measurement $\hat{z}$ and the covariance Z. Features with low importance are rejected as they are probably wrongly associated. Because of the global data association a misassociation will be propagated to every particle. So we have to make sure, that those features do not affect the weight of the particle by setting their importance to 1.
Because of the sensors used, multiple observations per time step are obtained. So the fastSLAM algorithm is modified by sampling the robot's pose once per time step and updating all observed landmarks for that pose. The weight of a particle i is then calculated by multiplying the importance of all features

$$w_i = \prod_{j=1}^{n} p_j$$

with the number of new observations n. Those features rejected by the best particle, i.e. the particle with the maximum weight after incorporating all observations of that time step, are added as new features both to the global kd-tree and all the particles.
After the update step the particles have to be resampled according to their weight. The particles with a higher weight are resampled more often than those with lower weight. During resampling, the distribution of the particles change and after resampling they are distributed according to the posterior.

5 Environment Reconstruction on Care-O-bot® 3

The 3-D environment modeling method used on Care-O-bot® 3 is meant to reconstruct single dynamic scenes of rather small environments out of colored point clouds. The purpose of the map is to model obstacles for navigation and manipulation. As presented in section 3, 2-D features from color images are used to associate the data of subsequent sensor readings. The fastSLAM algorithm from section 4 performs the map aggregation and corrects the robot position provided by laser scanner localization.
For many manipulation tasks a dense map is needed that makes it possible to create geometric objects. Feature-only maps cannpt guarantee a specific density. For this reason the raw point clouds are transformed into a consistent point cloud of the whole scene based on the feature map.

5.1 Intensity filtering

Care-O-bot® 3 is equipped with a SwissRanger 4000. Due to reflections and areas that are out of sensor range, intensity filtering according to section 2 has to take place. The intensity values range in the interval $(0, 30000)$ for our test environment (see figure 4) and good filtering results are obtained for $I_{th} = 1000$. Figure 2 shows a point cloud before and after filtering. It can be seen that most of the noise is removed. However, there are still points with incorrect depth values remaining. Those will be removed by the fastSLAM filter during data association.

Figure 2: Point cloud before (left) and after (right) intensity filtering.

5.2 Feature Extraction and Data Association

The colored point cloud is obtained by combining sensor data from a color camera and a time-of-flight camera. The two sensors are calibrated to each other and the point cloud is constructed by transforming the 2-D color data from the color camera into the TOF camera coordinate system. Those features masked by the intensity filtering step are removed and not used for data association. At an overlap of 70 % between two images 5-20 data associations are found (see figure 3). The rate of misassociations is smaller than 5 %.

Figure 3: Extracted features (blue circles) and correspondences (red lines) between two images.

5.3 System Model

The robot movement is estimated by the odometry motion model [12]. The robot only performs planar movements, so the robot state has three DOF, the position and orientation. The system model represents

$$p(s_t|u, s_{t-1})$$

with the robot state $s = (x, y, \phi)$ and the control u. The control of the odometry motion model is given by

$$u = \begin{pmatrix} x_{t-1} \\ x_t \end{pmatrix}.$$

Figure 4: Image of the kitchen environment (left), feature point map (middle), raw point cloud map (right)

As Care-O-bot® 3 relies on laser range finders for localization, that information is used instead of real odometry data. Nevertheless, the pose output of the localization component can be considered as odometry information.
During simulation, additional noise is added to the pose estimate. The covariance of the Gaussian noise

$$N = \begin{bmatrix} 0.0001 & 0 & 0 \\ 0 & 0.0001 & 0 \\ 0 & 0 & 0.00029 \end{bmatrix}$$

is found heuristically. As can be seen, the position uncertainty for our system is rather small due to laser-based localization instead of odometry measurements.

5.4 Measurement Model

As measurement model a feature-based model as introduced in [12] is used. This model basically reflects the transformation of a feature point from global to robot coordinates. This is a coarse modeling, but sufficient for our needs and easy to implement.
With the robot position s and the landmark position Θ the measurement z, i.e. the landmark position in robot coordinates can be calculated by

$$\begin{pmatrix} z_x \\ z_y \end{pmatrix} = \begin{pmatrix} \Theta_x \cdot \cos s_\phi - \Theta_y \cdot \sin s_\phi + s_x \\ \Theta_x \cdot \sin s_\phi + \Theta_y \cdot \cos s_\phi + s_y \end{pmatrix}.$$

Since the robot only moves planarly, the height of the landmarks is not incorporated in the measurement update. This is the reason why the robot pose does not provide information in that coordinate.
This modeling is rather crude because it does not take into account the physical measurement process. Because of that, the covariance matrix R cannot be determined analytically. So we have to identify this matrix by experiments. The main uncertainty comes from the time-of-flight camera. Therefore, images of a plain wall are taken from several distances and compared to the manually measured distances. With that it is possible to identify the mean and the variance of measured values which yield

$$R = \begin{bmatrix} 0.0025 & 0 \\ 0 & 0.0025 \end{bmatrix}$$

as covariance.

5.5 Map Aggregation

The output of the fastSLAM algorithm is a 3-D feature point map. Depending on the environment structure (i.e. the number of detected features), the density of the map can change. For some tasks the density of a sole feature point map can be sufficient. But for accurate manipulation in populated environments or as a base for the identification of geometric objects a raw point cloud map is a better input. Using the corrected robot position the raw point clouds can be transformed and merged into a single point cloud of the scene. Figure 4 shows the feature map and the raw point cloud map of a kitchen environment. The data is acquired in a single sweep and the head is not moved during data acquisition which yields a map that does not cover the full height of the kitchen.

5.6 Experimental Results

Practical experiments yield results concerning performance, data association and map aggregation. A test scenario is created by rotating the sensor head by 45° in steps of 5° and performing a measurement after each movement step. The rotational movement is done back and forth several times. The expectation is that the number of newly observed features decreases with each sweep over the scene. Furthermore, the error of the estimated robot position should be bounded.
On the whole, a sequence of 46 pictures is taken. With a SURF detection threshold of 500, in total 3625 SURF features are extracted. The threshold for data association is set to a descriptor error of 0.1 and 3308 data associations are found. During filter update mismatches are rejected. After this step, 2466 correct data associations remain, which is about 75 % of all observed features. Figure 5 shows the number of features in the map over the rotation angle. It can be seen that the number of features in the map only grows slightly after the first rotation direction change and is finally bounded. The increase of features can be explained by illumination change during the experiment and new features near the image boundaries at 0° and 45° due to inaccurate sensor movement. The estimated rotation angle of the robot is shown in brackets on the x-axis. The

maximum error is 0.5° and does not grow over time.

Figure 5: Number of features during a rotational movement of the robot.

6 CONCLUSIONS AND FUTURE WORKS

The paper proposed a method that is capable of reconstructing environment maps in 3-D. The method relies on 2-D features extracted from color camera images and transformed into 3-D features using a time-of-flight camera or stereo-vision. The algorithm was evaluated on Care-O-bot® 3 and proved to be able to handle given tasks.
In the area of environment modeling there are some improvements to be done in order to increase performance of the algorithm. As the kd-tree that holds all features grows during runtime and has to be rebuilt in every time step the update cycle of the algorithm slows down significantly for larger scenes. This can be tackled by hierarchical kd-trees. The idea is to group the features according to their position in the environment. Therefore, only the subtree belonging to the currently observed area has to be rebuilt.
At the moment, only feature or raw data point clouds can be constructed. Feature-only may not be dense enough for rendering obstacles in manipulation tasks, depending on the environment. The raw data point clouds contain many duplicate points due to overlapping scans. The map quality could be improved by introducing a resampling step that removes duplicate values and generate a point cloud with equally distributed points.

References

[1] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit: "FastSLAM: A factored solution to the simultaneous localization and mapping problem." *In: Proceedings of the AAAI National Conference on Artificial Intelligence*, pp. 593-598, 2002.

[2] A. Nüchter et. al.: "3D Robotic Mapping." *Springer Tracts in Advanced Robotics (STAR)*, ISBN 978-3540898832, 210 pages, Springer Verlag, 2009.

[3] A. Oyama, K. Konolige et. al.: "Come on in, our community is wide open for Robotics research!" *RSJ*, 2009.

[4] C. Parlitz, M. Hägele, P. Klein, J. Seifert and K. Dautenhahn: "Care-O-bot 3 - Rationale for human-robot interaction design." *In: International Federation of Robotics u.a.: ISR 2008 : 39th International Symposium on Robotics*, pp. 275-280, Seoul, Korea, 2008.

[5] M. Keller, A. Kolb: "Real-time simulation of time-of-flight sensors." *Simulation Modelling Practice and Theory*, Vol. 17, Issue 5, May 2009, pp. 967-978, ISSN 1569-190X, DOI: 10.1016/j.simpat.2009.03.004.

[6] H. Bay, A. Ess, T. Tuytelaars and L. Van Gool: "SURF: Speeded Up Robust Features." *In: Computer Vision and Image Understanding (CVIU)*, Vol. 110, No. 3, pp. 346-359, 2008.

[7] D. G. Lowe: "Distinctive Image Features from Scale-Invariant Keypoints." *In: International Journal of Computer Vision*, Vol. 60, No. 2, pp. 91-110, 2004.

[8] J. Kubacki and K. Pfeiffer: "Using Range Imaging Sensors with Color Imaging Sensors in Cognitive Robot Companions: A New and Simple Calibration Technique Based on Particle Swarm Optimization". *In: Ingensand, Hilmar (Publ.) u.a.; ETH Zürich: 1st Range Imaging Research Day: Proceedings*, pp. 43-57, Zürich, Switzerland, 2005.

[9] M. Özuysal, M. Calonder, V. Lepetit and P. Fua: "Fast Keypoint Recognition using Random Ferns." *In: IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 32, No. 3, pp. 448 - 461, 2010.

[10] M. Montemerlo, S. Thrun, D. Koller and B. Wegbreit: "FastSLAM 2.0: An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping that Provably Converges." *In: Proc. of the Int. Conf. on Artificial Intelligence (IJCAI)*, pp. 1151-1156, 2003.

[11] P. Neubert, N. Sünderhauf, P. Protzel: "FastSLAM using SURF Features: An Efficient Implementation and Practical Experiences." *In: Proc. of the International Conference on Intelligent and Autonomous Systems (IAV)*, Toulouse, France.

[12] S. Thrun, W. Burgard, and D. Fox: Probabilistic Robotics. MIT Press, Cambridge, MA, 2005.