

OctoSLAM: A 3D Mapping Approach to Situational Awareness of Unmanned Aerial Vehicles

(Demonstration)

Joscha Fossel, Daniel Hennes, Sjriek Alers, Daniel Claes, Karl Tuyls
Department of Knowledge Engineering, Maastricht University
P.O. Box 616, 6200MD Maastricht, The Netherlands
{j.fossel,daniel.hennes,sjriek.alers,daniel.claes,k.tuyls}@maastrichtuniversity.nl

ABSTRACT

Unmanned aerial vehicles (UAVs) have recently become widely available to the research community. A common vision is that such (semi-)autonomous airborne agents can be beneficial in numerous scenarios, e.g. urban search and rescue. However, when deploying computationally restricted UAVs in these real life scenarios, various challenges from multiple research domains arise. These include situational awareness, controlling, planning, and learning.

The focus of this demonstration is on situational awareness of agents capable of 6D motion, in particular UAVs. We propose the integration of 2D laser range finder, altitude, and attitude sensor data to compose 3D maps of the environment. Experiments show significant improvement in the localization and representation accuracy over current 2D map methods.

Categories and Subject Descriptors

I.2.9 [Artificial Intelligence]: Robotics

General Terms

Algorithms, Experimentation

Keywords

Situational Awareness, UAV, Quadrotor, SLAM

INTRODUCTION AND MOTIVATION

Multi-rotor Unmanned Aerial Vehicles (UAVs) are popular yet complex systems. UAVs generate a lot of interest due to their broad scope of applications. The autonomous exploration of large, inaccessible or hazardous environments are prominent examples. A prerequisite to any such autonomous behavior is the ability of the agent to perceive environmental elements and to localize itself.

Situational awareness of UAVs is commonly tackled by either time of flight (e.g. laser range finder) or visual (e.g. depth cameras) simultaneous localization and mapping (SLAM). While visual sensors may provide 3D data, they are also lighting dependent, have a lower range and less precision than time of flight sensors. Visual SLAM also tends to be more computationally expensive than time of flight SLAM, which can be crucial on airborne robots which typically have limited payload and processing capabilities. Therefore,

Appears in: *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2013)*, Ito, Jonker, Gini, and Shehory (eds.), May, 6–10, 2013, Saint Paul, Minnesota, USA.

Copyright © 2013, International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

we opt to use time of flight sensors. However, 3D time of flight sensors are heavy, expensive and have a high power consumption. Thus, only 2D time of flight sensors have been used on UAVs for SLAM so far. The established practice is to down-project the laser range data to 2D maps. This assumes that objects look the same regardless of the observation height. On the one hand, if this assumption holds, 2D map SLAM, e.g. particle filter based SLAM, is sufficient. On the other hand, if the assumption does not hold, using 2D maps in the UAV domain becomes impracticable. Therefore, we develop a 2D laser range finder based SLAM algorithm that operates on a 3D representation of the environment.

Next, we summarize our approach, called OctoSLAM, which advances the state-of-the-art low resource requiring SLAM frameworks for agents exhibiting 6D motion. We use the Robot Operating System to implement OctoSLAM.

PROPOSED SLAM APPROACH

For 3D map SLAM we combine and extend both Octomap [2], an octree representation of the environment, and Hector SLAM [1], an algorithm for fast online learning of occupancy grid maps.

The 6D pose of the agent is determined by both localization and direct sensor input. The latter is used for roll, pitch and altitude. The remaining three dimensions, i.e. translation in x , y direction and rotation around the yaw axis (γ), are tracked through localization. At time t the position is given by $T_t = (x_t, y_t, \gamma_t)$. The algorithm iteratively computes the most recent pose change $\Delta T = T_t - T_{t-1} = (\Delta x, \Delta y, \Delta \gamma)'$. The laser range finder returns a vector of scan endpoints $D = (d_0, \dots, d_N)$ in polar coordinates. Via roll, pitch, attitude, and pose estimate, scan endpoints d_i are transformed to the Cartesian coordinates of the map representation, i.e. to 3D. We denote this transform by $T_t \otimes d_i$. Hector SLAM uses map gradients ∇M to align the scan D with the current map representation. OctoSLAM computes interpolated map gradients based on occupancy probabilities of the target and surrounding nodes of the Octomap representation. The pose change ΔT is computed by a Gauss-Newton equation. For ease of notation we write T instead of T_t in the following.

$$\Delta T = H^{-1} \sum_{i=1}^N \left[\nabla M(T \otimes d_i) \frac{\partial(T \otimes d_i)}{\partial T} \right]' [1 - M(T \otimes d_i)].$$

The Hessian H is computed as follows:

$$H = \left[\nabla M(T \otimes d_i) \frac{\partial(T \otimes d_i)}{\partial T} \right]' \left[\nabla M(T \otimes d_i) \frac{\partial(T \otimes d_i)}{\partial T} \right],$$

where $\nabla M(d_i \otimes T) = (mv_i, dy_i, dx_i)'$ are the bilinearly interpolated map value gradients. The partial derivative $\frac{\partial(T \otimes d_i)}{\partial T}$ is com-

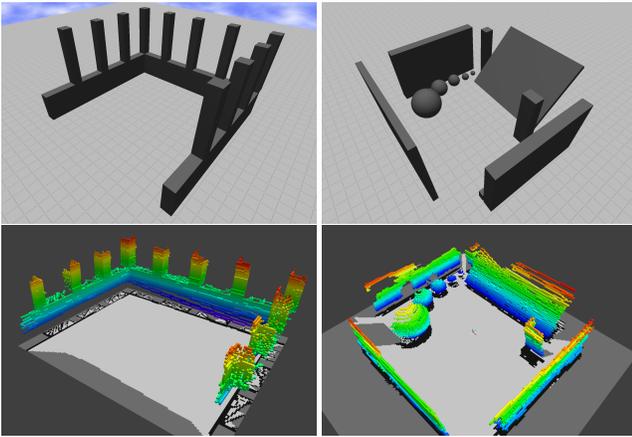


Figure 1: In the first column the corridor, and in the second column the tilted wall world is shown. The first row shows the according world in the simulator, and the second row shows examples for OctoSLAM generated maps.

puted as:

$$\frac{\partial(T \otimes d_i)}{\partial T} = \begin{pmatrix} 1 & 0 & -\sin(\gamma)d_i^x & -\cos(\gamma)d_i^y \\ 0 & 1 & \cos(\gamma)d_i^x & -\sin(\gamma)d_i^y \end{pmatrix}.$$

Furthermore, if a scan endpoint is located in a so far unknown part of the map, the voxel size used for determining the interpolated map value and gradients is increased. Since parent nodes in octrees aggregate the information stored in the according child nodes, this operation neither increases memory nor computational requirements. Using this heuristic the probability of getting stuck in a local minima due to a insufficient scan to map fit is reduced.

RESULTS

We test OctoSLAM in both simulation and real environments. The sensor readings of every trial are recorded and used as input for the different SLAM approaches.

Figure 1 shows two of the simulated worlds used for evaluating the SLAM performance. Additionally, examples for corresponding maps generated by OctoSLAM can be seen. Figure 2 presents the results for the simulated experiments. As error measure we use the euclidean distance between true and localized pose. Each graph shows the average localization root mean squared error (RMSE) (middle line), the 95% confidence interval (inner box), the standard deviation (outer box) and the trial RMSEs (points). Threshold mapping refers to updating the map only if the pose has changed by a certain amount and is commonly employed when using 2D maps. Constant mapping in contrast refers to updating the map with every obtained sensor reading, which is necessary to build a 3D map from 2D laser range finder data. Using a 3D map for SLAM yields significantly smaller errors in both simulated worlds. Figure 3 shows the real robot test environment and the resulting maps for 2D map and 3D map SLAM. To allow for easier comparison, the internally used 2D map is not shown, but instead a separate 3D map that is additionally recorded. The 3D map generated by 2D map SLAM is severely less precise than that generated by 3D map SLAM.

DEMONSTRATION

We demonstrate OctoSLAM in simulation and in a real-world setting. In simulation, OctoSLAM is performed on a manually controlled quadrotor exploring various unknown environments. In

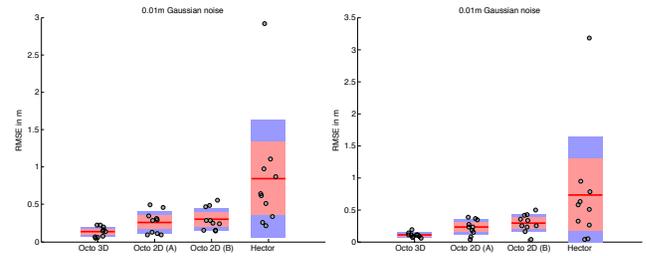


Figure 2: On the left the SLAM results for the corridor world, and on the right for the tilted wall world are shown. (A) refers to threshold mapping, (B) to constant mapping.

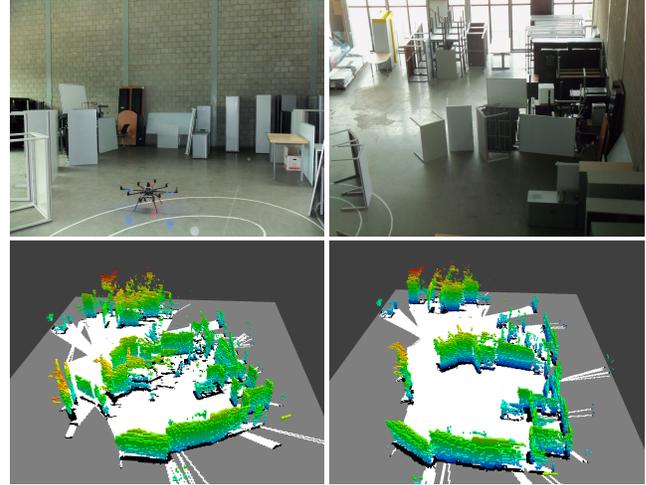


Figure 3: This figure shows the real robot test environment from two different angles in the first row. In the second row the resulting maps are shown - for 2D map SLAM on the left, and for 3D map SLAM on the right.

the real-world setting, our approach is demonstrated on a UAV equipped with a Hokuyo URG laser range finder, Pololu IMU and Parallax ultrasonic sensor. If the venue is not suited for UAV flight, we demonstrate our approach on a handheld device instead. A demo video is available at: <http://www.goo.gl/aB3dy>

CONCLUSIONS

We have summarized a new situational awareness approach for autonomous UAV operation that can be used on any sensor wise appropriate agent exhibiting 6D motion. The presented experimental results clearly demonstrate the effectiveness and applicability of OctoSLAM in various unstructured environments.

REFERENCES

- [1] S. Kohlbrecher, J. Meyer, O. von Stryk, and U. Klingauf. A Flexible and Scalable SLAM System with Full 3D Motion Estimation. In *Proceedings of IEEE International Symposium on Safety, Security and Rescue Robotics*. IEEE, 2011.
- [2] K. M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard. Octomap: A probabilistic, flexible, and compact 3d map representation for robotic systems. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2010.