Particle Filter Based SLAM for Forestry Robot

Sylvain Geiser[*]
Department of Interdisciplinary Informatics, Graduate School of Computer Science and Systems Engineering, Kyushu Institute of Technology
680-4, Kawazu, Iizuka-City, Fukuoka, 820-8502, Japan

Sakmongkon Chumkamon
Department of Mechanical Information Science and Technology, Kyushu Institute of Technology
680-4, Kawazu, Iizuka-City, Fukuoka, 820-8502, Japan

Ayumu Tominaga
National Institute of Technology, Kitakyushu College
5-20-1 Shii, Kokuraminamiku, Kitakyushu, Fukuoka, 802-0985 Japan

Takumi Tomokawa
Department of Mechanical Information Science and Technology, Kyushu Institute of Technology
680-4, Kawazu, Iizuka-City, Fukuoka, 820-8502, Japan

Eiji Hayashi
Department of Mechanical Information Science and Technology, Kyushu Institute of Technology
680-4, Kawazu, Iizuka-City, Fukuoka, 820-8502, Japan
E-mail: geiser.nathan-sylvain@mail.kyutech.jp, m-san@mmcs.mse.kyutech.ac.jp, tominaga@kct.ac.jp, tomokawa.takumi163@mail.kyutech.jp, haya@.mse.kyutech.ac.jp,
http://www.kyutech.ac.jp/

Abstract

In Japan, the forestry workforce is dramatically declining. Therefore, field robots are investigated to replace humans for dangerous actions. Task execution with such mobile robots requires localization and mapping. This research focuses on online SLAM implemented on SOMA forestry robot developed at Hayashi Laboratory. In this approach, the core algorithm is a Rao-Blackwellized particle filter. A realistic simulation has been build using Gazebo and the results of first experiments speak for real-time capability.

Keywords: Field Robot, online SLAM, Forestry, Particle Filter

1. Introduction

Japanese forestry workers have a more and more advanced age. Consequently, because little youths want to apply for a job in this sector, the number of people in the field is shrinking. At the same time, the need of forest management is increasing. Owing to all this factors, in addition to hazardous nature of some forestry tasks, introducing robots on the field could be beneficial. One prototype for this purpose is currently designed at

[*]7 rue Peyron, Vienne (38200), France

Hayashi Laboratory. This robot, named SOMA,[1] has been built from an All-Terrain Vehicle (ATV) and many sensors like odometers, GPS (Global Positioning System), IMU (Inertial Measurement Unit), RGB-D cameras and lidar have been fixed to it.

Every mobile robot – including forestry ones – requires some navigation system to be able to execute its tasks and Simultaneous Localization and Mapping (SLAM) is an essential part of it. Unless most applications where full SLAM is first solved only for mapping and then localization is performed within the previously built map, this research is investigating online SLAM for forestry robots. This approach is motivated by the fact that building a map in advance demands time. Furthermore, it needs to be done every time the environment where the robot evolves changes. As a result, it enables computation savings only when the world stays identical for a long time. However, forests are constantly changing because of trees being planted, growing and being felled. In addition, the robot can be led to work in different areas. This point explains the choice of solving online SLAM for this specific application.

Among various algorithms available, FastSLAM was chosen because of its multimodal beliefs management ability, inherent to its use of a particle filter.[2]

This article begins with the description of the approach and how simulation experiments were conducted. Then, the obtained results are detailed and analyzed, before conclusion and opening on future research being drawn.

2. FastSLAM for Forestry Robot

The FastSLAM algorithm is based on a Rao-Blackwellized particle filter in which a particle represents a joint hypothesis of the pose of the robot and the map through the positions of landmarks.[2] The uncertainty of the map is represented by associating a Kalman filter to each feature in it. Thus, each particle contains a weight, a pose and a collection of Kalman filters depicting the environment.

Different parts of the algorithm need to be specified according to the forestry context. First, since trees are very common and easily distinguishable in a forest, there were chosen as landmarks to which the environment is mapped.

A particle filter being a variety of Bayes filter, it also requires motion and observation models to be implemented.

2.1. *Motion*

The motion model is based on the odometry captured by the rotary encoders mounted on the wheels of the robot. Each movement is decomposed in three parts: a first rotation, a translation and a second rotation, and linear gaussian noise is assumed for each of them.

2.2. *Observation*

With regard to observation, the sole sensor used is the lidar on top of the robot and thereby a range-bearing model was chosen. The coordinates of the trees are determined by extracting the centers of 3D clusters made from lidar pointcloud, after having clipped the latter in height to remove ground and foliage and keep only tree trunks (Fig. 1). Assuming linear gaussian noise for individual readings of the lidar, these coordinates can be proved to be also corrupted with gaussian noise.

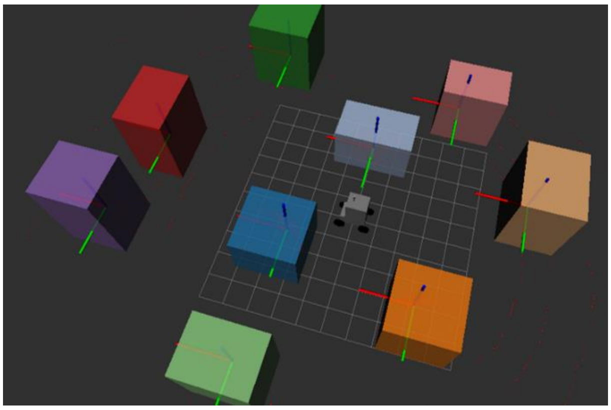

Fig. 1. Clusters made from lidar pointcloud

2.3. *Correspondences and map management*

Another important part of every SLAM solver is how to get correspondences between observations and features in the map being built. The largely used maximum likelihood approach was chosen. In FastSLAM, correspondences are determined per particle and not for the whole particle set, enabling some diversity of data association.

2.1.1. *Multiple observations*

Multiple observations are often split into individual ones and processed sequentially. However, it can lead to wrong fusions of landmarks, because two observations generated by two different landmarks can be assigned to the same feature in the map. On account of this issue, we here try to handle multiple observations at once thanks to the Gale-Shapley algorithm.[3]

The likelihood of each feature in the lidar visibility scope is computed for each observation and a list of features ordered by decreasing probability is made for the latter. Then, each observation is appaired with the first feature in its list. If more than one observation is linked with the same feature, only the observation with the highest likelihood keeps it and the second-ranked ones are attributed to the others. This step is repeated until each observation is appaired with a different feature. When the likelihood of an attributed feature is less than a threshold or when there are no more features in the list to continue the process, a new feature is created for the corresponding observation.

2.1.1. *Features deletion*

Features can also be subject to deletion when it appears that they do not correspond to a tree anymore. This action is realized along with correspondences establishment and the same threshold is used. Of course, it is not applied to features which are not considered to be in the lidar visibility scope.

2.4. *Particles handling*

Particle filters are often subject to what is named particle deprivation, that is the decrease of the diversity of the particle set over time. Therefore, resampling is used to counteract this drawback. In addition, resampled particles are slightly randomized by applying gaussian noise with the same order of magnitude than motion noise.

3. Experiments

For simulation purposes, a model of SOMA robot has been developed in Rviz and Gazebo using ROS, along with a realistic forest environment. The latter is made of a ground and 16 pine trees and measures 30 by 30 meters (Fig. 2).

Fig. 2. Robot model in Gazebo environment

In order to simulate motor noise, a ROS node was introduced between the steering controller and Gazebo. This node adds linear gaussian noise to linear and angular velocity commands before publishing them to a new topic to which Gazebo subscribes (Fig. 3).

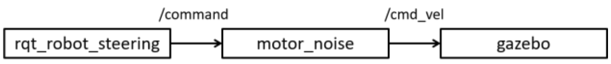

Fig. 3. Graph of motion nodes and topics

Different experiments have been conducted in order to determine the effect of the number of particles and the number of trees on the overall precision and update rate of the implementation. For each of them, the robot is going straight from the left to the right in the forest environment.

4. Discussion

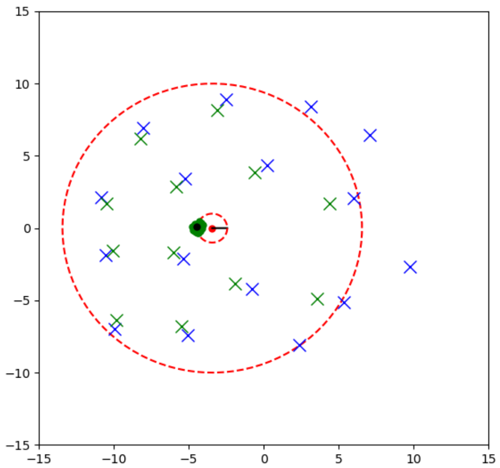

Fig. 4. Display of simulation (red point is real position, black line in front of red point is real heading, space between the two dotted red circles is lidar visibility scope, blue crosses are trees, green points are positions of particles, black point is position of the most probable particle, green crosses are features in the map of the most probable particle)

With realistic motion and observation noises and 100 particles, the distance between real and estimated poses at the end of the simulation is about 2.39 m, and the average distance between trees and associated features is 0.81 m (Fig. 4). Five trees over 16 are represented by multiple features and one feature does not correspond to any tree. In addition, the update rate in this case is about 1 Hz.

Overall, the robot is well tracked along its path. However, we can notice that the pose estimation is increasingly late compared to the real pose. This systematic shift can be explained by two factors. First, because this effect is observed even without motion noise, we can suppose odometry to be captured late. However, the main cause seems to be the imprecision of the computation of the coordinates of the trees. Indeed, the latter is based on the clusters made from lidar pointcloud. The centers of these do not coincide with the centers of trunks, because only a part of each tree is seen by the robot at one time. As a consequence, the center of each cluster is changing a lot as the robot is moving around the corresponding tree (Fig. 5).

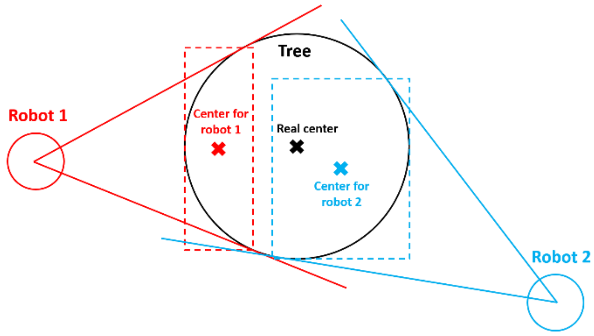

Fig. 5. Centers of clusters depending on the pose of the robot

Thus, depending on the ratio between motion and observation noises, the pose estimation can be overly updated according to this modification of the coordinates of the features, and induce a shift between real and estimated poses. This is why the use of circle pattern recognition should be used in each cluster to get the right coordinates of trees in future research.

4.1. *Effect of the number of particles*

The number of particles does not seem to have a very significant effect on the overall precision of the implementation. Even 5 particles are sufficient to get quite accurate final pose and map. As expected, the update rate decreases with the number of particles. Finally, adding more particles do not solve the systematic shift issue.

4.2. *Effect of the number of trees*

The closer the robot goes next to the trees, the more important the effect of systematic shift, because less of the tree is seen at one time. Then, the amplitude of change of the coordinates of the center of the cluster is larger in this case. Thereby, a higher density of trees leads to a lower precision. Moreover, as already known in the context of SLAM, more landmarks induce more wrong data association. Finally, the total number of trees does not affect the update rate of the implementation, because only features in the lidar visibility scope are updated. Nevertheless, the denser the forest, the lower the update rate.

4.3. *Future research*

As previously mentioned, a circle pattern recognition should be used to compute the coordinates of the trees from the lidar pointcloud, in order to remove the described systematic shift between the real and estimated poses.

In addition, resampled particles are here randomized to prevent particles deprivation, but more advanced methods exist such as mixture MCL, which is much more efficient in particular when motion noise dominates observation noise.

Finally, more experiments including real ones should be conducted to completely evaluate the approach described in this paper.

References

1. N. Takegami, E. Hayashi, R. Fujisawa, “Environment map generation in forest using field robot”, Proceedings of International Symposium on Applied Science 2019
2. M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit, “FastSLAM: A Factored Solution to the SLAM Problem”, 2002
3. D. Gale, L. S. Shapley, “College Admissions and the Stability of Marriage”, The American Mathematical Monthly, Vol. 69, No. 1 (Jan., 1962), pp. 9-15

Authors Introduction

Mr. Sylvain Geiser

He studied Engineering and Computer Science in Ecole des Mines de Nancy, France, between September 2018 and June 2020. He is currently a Master student at Kyushu Institute of Technology, Japan, and conducts research at Hayashi Laboratory.

Dr. Sakmongkon Chumkamon

Dr. Sakmongkon Chumkamon received Doctor of Engineering degree from Kyushu Institute of Technology in 2017. He was a postdoctoral researcher at Guangdong University of Technology in 2017-2019. Presently he is a postdoctoral researcher in Kyushu Institute of Technology since 2019. His research interests include factory automation robots and social robots.

Prof. Ayumu Tominaga

Prof. Ayumu Tominaga is a professor in the Department of Intelligent and Control Systems at Kyushu Institute of Technology. He received the Ph.D. (Dr. Eng.) degree from Kyushu Institute of Technology in 2021. His research interests include Intelligent mechanics, Mechanical systems and Perceptual information processing.

Mr. Takumi Tomokawa

He received bachelor degree in Engineering in 2021 from mechanical system engineering, Kyushu Institute of Technology in Japan. He is aquiring the Master degree in Kyushu Institute of Technology.

Prof. Eiji Hayashi

Prof. Eiji Hayashi is a professor in the Department of Intelligent and Control Systems at Kyushu Institute of Technology. He received the Ph.D. (Dr. Eng.) degree from Waseda University in 1996. His research interests include Intelligent mechanics, Mechanical systems and Perceptual information processing. He is a member of The Institute of Electrical and Electronics Engineers (IEEE) and The Japan Society of Mechanical Engineers (JSME).