# Implementation of Autonomous Navigation Using a Mobile Robot Indoor

**Sung Woo Noh, Dong Jin Seo, Tae Gyun Kim, Seong Dae Jeong and Kwang Jin Kim**

**Abstract** This paper describes an implementation of autonomous navigation of a mobile robot indoors. The implementation includes map building, path planning, localization, local path planning and obstacle avoidance. ICP(Iterative closest point) is employed to build grid based map using scanned range data. Dijkstra algorithm plans the shortest distance path from a start position to a goal point. Particle filter estimates the robot position and orientation using the scanned range data. Elastic force is used for local path planning and obstacle avoidance towards a goal position. The algorithms are combined for autonomous navigation in a work area, which comprises indoor environments with different types. The experiments show that the proposed method works well for safe autonomous navigation.

**Keywords** Autonomous navigation · Localization · Map building · Path planning · Obstacle avoidance

## 1 Introduction

Autonomous navigation is one of the most fundamental functions for practical use of a mobile robot. There have been abundant researches on autonomous navigation indoors and many of them are implemented for practical use. However, still these algorithms are not easily accessible to users for practical use.

S.W. Noh(✉) · D.J. Seo · T.G. Kim · S.D. Jeong
Sinmyeong Urban Information Co.LTD,
333 Cheomdan Kwagiro Bukgu, Gwangju 500-706, Korea
e-mail: nswking0212@gmail.com

D.J. Seo · T.G. Kim · S.D. Jeong · K.J. Kim
Gwangju Technopark Robot Center,
333 Cheomdan Kwagiro Bukgu, Gwangju 500-706, Korea
e-mail: kjkim@gjtp.or.kr





D.-S. Park et al. (eds.), *Advances in Computer Science and Ubiquitous Computing*,
Lecture Notes in Electrical Engineering 373,
DOI: 10.1007/978-981-10-0281-6_106

Sometimes, users feel difficulty adapting algorithm components for navigation implementation. As an example, though there are many algorithms for obstacle avoidance, it is not easy to incorporate the algorithm into navigation software for practical implementation. This paper reports one of the successful implementation of the indoor navigation which is practically feasible in the respect that it combines map building, path planning, localization, local path planning and obstacle avoidance, and path tracking in a coherent manner[1]. Section 2 describes each algorithm component for integrated navigation implementation and section 3 shows an example of navigation in a building using the proposed method. Section 4 concludes the research results.

## 2 Element Technology for Navigation

### *2.1 Map Building*

The map required for the navigation is built before the navigation and is provided to the motion algorithms. For the map building, ICP (Iterative closest point) algorithm is used. The ICP provides grid based map using the scanned range data gathered by a Laser range finder (LRF) mounted on a mobile robot which is driven by an operator. Fig. 1 shows a grid map of a building. The map represents an environment of 100m length with 40m width work area of a floor by the grid resolution of 10cm×10cm.

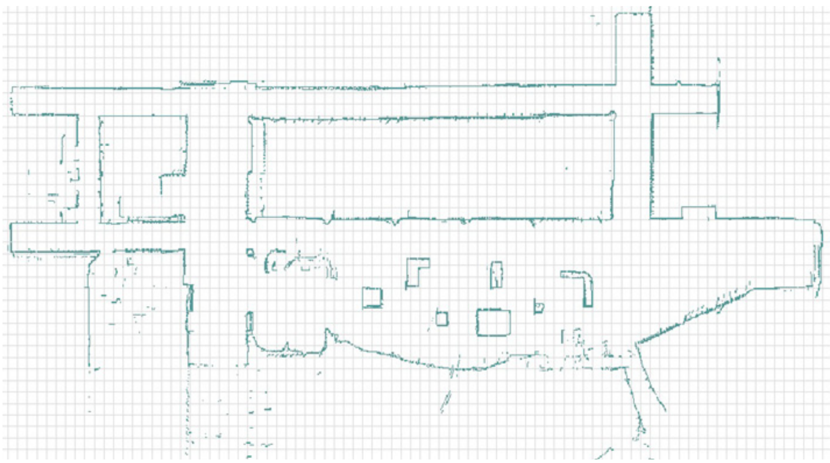

**Fig. 1** A grid map of the work area of 100m×40m

### *2.2 Localization*

A particle filter based method, called the MCL (Monte Carlo Localization) is used for estimation of robot location[2]. As with usual Bayes filtering estimation methods, the MCL consists of prediction and correction steps. Table 1 shows pseudo code of the MCL algorithm. Every particle of the method represents a possible location of the robot. Line 3 of the Table 1 depicts the prediction step which calculates a possible location of a particle based on dead reckoning. Line 4 assigns belief to each predicted particle using measured sensor data. In our research, the data for belief calculation is the scanned range data to the wall from the LRF on board the robot. Lines 6 to 9 resamples possible locations form the predicted particles to provide corrected possible locations. Therefore, these lines correspond to the correction step and complete an iteration of the MCL.

**Table 1** Localization method using particle filter

$Algorithm\ MCL(X_{t-1}, u_t, z_t, m)$

$\{$

$\quad \bar{X}_t = X_t = \phi$

$\quad for\ \ m = 1\ to\ M\ \ do$

$\quad\quad x_t^{[m]} = sample\_motion\_model(u_t, x_{t-1}^{[m]})$

$\quad\quad \omega_t^{[m]} = measurement\_model(z_t, x_t^{[m]}, m)$

$\quad endfor$

$\quad for\ \ m = 1\ to\ M\ \ do$

$\quad\quad draw\ i\ with\ probability \propto \omega_t^{[i]}$

$\quad\quad add\ \ x_t^{[i]}\ to\ X_t$

$\quad endfor$

$\quad return\ \ X_t$

$\}$

## 2.3 Path Planning and Obstacle Avoidance

The global path from a starting location to a goal location is planned using Dijkstra algorithm, which yields the shortest path through way points in the map[3].

Once the way points from a starting location to a goal location are given, local path planning and tracking is needed to move the robot through the way points while avoiding local obstacles detected during the navigation. Applying virtual elastic force and repulsive force to the path segment from the robot to the adjacent way point produces smooth and collision free local path to the way point. The following equation is used for the repulsive force from the obstacles.

$$V_{rep}\left({}^{j}P_i\right) = \begin{cases} \frac{1}{2} K_r (d_r - d({}^{j}P_i))^2, & if\ d({}^{j}P_i) < d_r \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

$${}^{r}f_i = -\nabla V_{rep}({}^{j}P_i) = K_r (d_r - d({}^{j}P_i)) \frac{\vec{d}}{d} \tag{2}$$

Eq. (1) describes repulsive force field and Eq. (2) calculates the repulsive force ${}^{r}f_i$ from obstacles to the robot located at ${}^{j}P_i$. In these equations, $d({}^{j}P_i)$ is the shortest distance between ${}^{j}P_i$ and obstacles, $d_r$ is the virtual distance of repulsive field influence, and $K_r$ is the control parameter of the repulsive force. Elastic force exerted to the point ${}^{j}P_i$ which is on the local path is given by the Eq. (3) [4].

$$ {}^{e}f_i = K_c\left(\frac{d_i^{j-1}}{d_i^{j-1}+d_i^{j}}({}^{j+1}P_i - {}^{j-1}P_i) - ({}^{j}P_i - {}^{j-1}P_i)\right) \tag{3} $$

$d_j^i$ is the distance between ${}^{j}P_i$ and ${}^{j+1}P_i$, and the $K_c$ is the elastic force parameter. The force to the robot at ${}^{j}P_i$ is the sum of the repulsive force and elastic force. Fig. 2. shows collision avoidance of local path to the way point in a simulation

$$ {}^{j}F_i = {}^{r}f_i + {}^{e}f_i \tag{4} $$

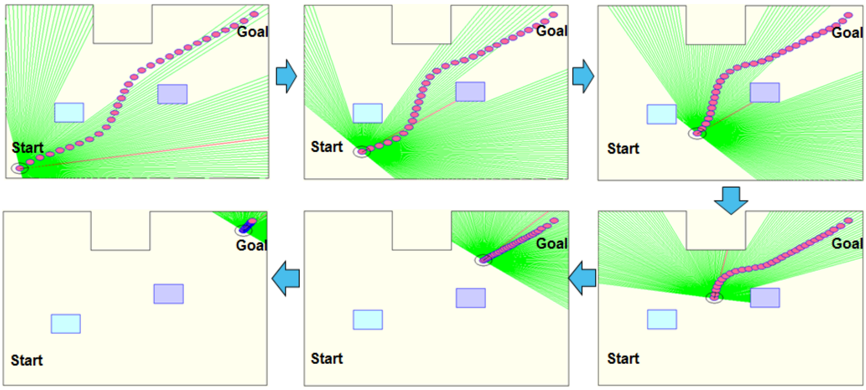


**Fig. 2** Obstacle avoidance process using the elastic force

## 2.4 Integration of the Algorithm Components

The method integrates the algorithm components in a coherent manner[5]. Map building algorithm provides map data to path planning, local path planning and tracking. Localization algorithm provides the robot location to local path planning and tracking. Planned global path is provided to local path planning and tracking. Detected obstacle is fed to local path planning and tracking. There algorithms are interconnected, and finally the local path planning and tracking produces command to the robot, that is, linear speed and angular speed of the robot motion.

# 3 Experiments

The method is implemented for navigation in a building of work range 100m×40m. The range sensor used for map building, localization, and obstacle detection is the LMS511 of Sick. The experiment uses a differential drive mobile robot. The maximum speed is set to be 1.0m/sec. Fig. 3 shows the navigation path and the path generated by Dijkstra algorithm. Fig. 4 shows an autonomous navigation trajectory.

The overall distance from the initial location to the goal location is 165m, and it takes 255sec for the travel, thus the average speed is 0.65m/sec. One of the corridors has the width of 2.2m and length 9.5m, and the other has the width 2.35m and the length 37m. The robot was able to avoid passers-by in the narrow corridors as well as in the lobby.

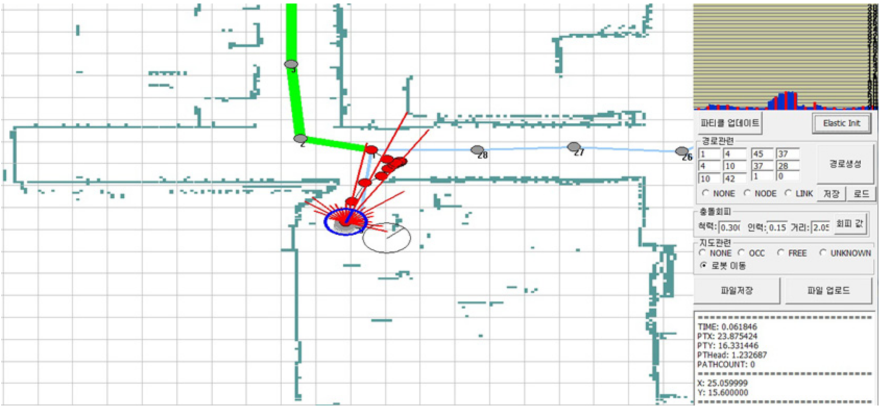


**Fig. 3** Trajectory of robot motion in the experiment.

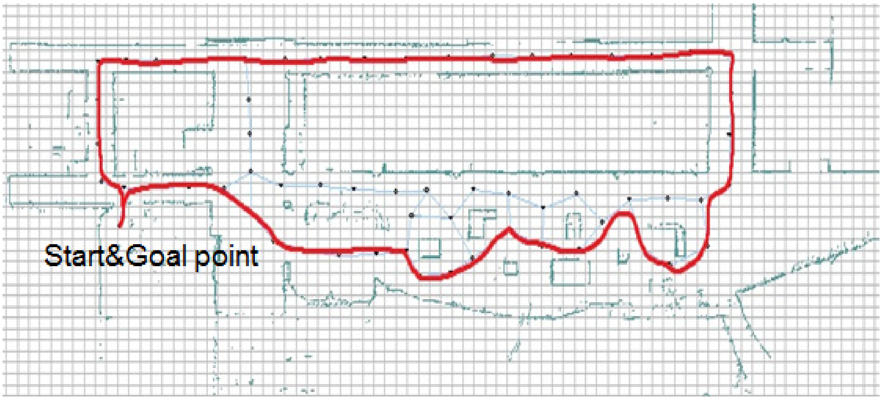


**Fig. 4** Trajectory of an autonomous navigation in the experiment.

# 4 Conclusion

The paper reports an implementation of autonomous navigation in indoor environment. It integrates several algorithms for mapping, planning, localization, and tracking, Through experiments, the method reveals reliable and robust navigation performance. There are other algorithm elements to be integrated to the proposed method such as map update and path re-planning. When the robot detects an obstacle which stationary, then the map should be updated to include the obstacle. Also, if the corridor along which the planned path lies is blocked by obstacles, the path should be re-planned globally.

**Acknowledgement** The research was supported by 'Software Convergence Technology Development Program', through the Ministry of Science, ICT and Future Planning(S0170-15-1079).

# References

1. Thrun, S.: Learning metric-topological maps for indoor mobile robot navigation. Artificial Intelligence **99**(1), 21–71 (1998)
2. Ko, N.Y., Kim, T.G., Noh, S.W.: Monte carlo localization of underwater robot using internal and external information. In: Proc. 2011 IEEE Asia-Pacific Services Computing Conference (APSCC), pp. 410–415, December 12-15, 2011
3. Tsourveloudis, N.C., Valavanis, K.P., Hebert, T.: Autonomous vehicle navigation utilizing electrostatic potential fields and fuzzy logic. IEEE Transactions on Robotics and Automation **17**(4), 490–497 (2001). 8
4. Seo, D.J., Ko, N.Y., Simmons, R.G.: An elastic force based on collision avoidance method and its application to motion coordination of multiple robots. International Journal of Computer Integrated Manufacturing **22**(8), 784–798 (2009)
5. Ko, N.Y., Noh, S.W., Moon, Y.S.: Implementing indoor navigation of a mobile robot. In: 2013 13th International Conference on Control, Automation and Systems (ICCAS), pp. 198–200. IEEE (2013)