

A Novel Detection Strategy for Kidnapped Robot Problem in Monte Carlo Localization Using Corridor-Type Map

Iksan Bukhori¹, Zool Hilmi Ismail², Tohru Namerikawa³

¹Study Program of Electrical Engineering, President University, Jalan Ki Hajar Dewantara 17550, Bekasi, Indonesia

²Department of Electronic Systems Engineering, Malaysia-Japan International Institute of Technology Universiti Teknologi Malaysia, Jalan Sultan Yahya Petra, 54100, Kuala Lumpur, Malaysia.

³Department of System Design Engineering, Keio University, 3-14-1 Hiyoshi, Kohoku-ku, Yokohama 223-8522, Japan.

Corresponding author: iksan.bukhori@president.ac.id

ABSTRACT

This paper proposed a novel method to detect the kidnapped robot problem event in Monte Carlo Localization in corridor-like map. The proposed method is aimed towards overcoming two common drawbacks from existing methods of detection, namely the inability to stay accurate across wide array of particles' convergence level, and a high dependency on the success of recovery process. This objective is achieved by combining the difference in particle's weight, maximum current weight, and difference in particles' standard deviation. The addition of these two parameters makes the proposed method to be superior to pure maximum current weight parameter for kidnapping detection. A series of simulation tests using corridor-like map are executed to test the claim. These simulations show that the proposed method outperforms the maximum current weight parameter in terms of accuracy, ability to detect kidnapping during early stage of localization, and independency towards the success of the re-localization process.

Keywords: Monte Carlo Localization, Kidnapping Detection, Difference of Weight, Difference of Standard Deviation, Early Kidnapping, Recovery Independency

I. INTRODUCTION

Kidnapped Robot Problem (KRP) in mobile robot localization is defined as a condition when the robot is instantly moved to other position arbitrarily without any prior given information during the operation of the robot [1], [7-9]. Kidnapped robot problem is considered one of the most difficult problem in Monte-Carlo Localization (MCL) [2]. This is due to the nature of particle filter used in MCL itself, where the convergence process of hypotheses (particles) causes an absence of particles in some areas of the map. The better the localization process, the more areas in the map lost the particles. This inevitably leads to a failure in re-localization if the robot is kidnapped to that area.

Kidnapped robot problem does not often happen in practice; however, it is often used to test the ability of algorithm to recover from global localization failures. Furthermore, the mechanical and sensor faults can lead to similar condition to kidnapping condition, thus the detection of this event can also be used as fault detection [6].

For decades, there have been several approaches in solving kidnapped robot problem. Some solutions are based on visual recognition, such as the ones found in [4] and [5]. These approaches, however, are limited to the robot with visual-based sensor, such as camera. Some other approaches are more flexible by using the intrinsic parameters of the MCL itself. Augmented MCL proposed in [1] and MCL with mixture distributions [1], [8], and [18] are some examples of this category.

In Augmented MCL, random particles are injected in each iteration such that the possibility of particles' absence in kidnapping destination area is reduced. These random particles are drawn from either uniform distribution over pose space, or the posterior of the measurement. MCL with mixture proposal distribution combines regular MCL sampling with its dual.

Despite its flexibility, the former two methods do not clearly draw a line between detection and recovery of kidnapping. This creates a problem when the concern is not only in the re-localization, but also the needs to know when the kidnapping really happens, such as in fault detection.

Other solutions which also depend on intrinsic parameters of MCL can be found in [2] and [3]. Zhang

et.al in [2] uses maximum weight of current particle set as the parameter to detect the kidnapping event. Yi, C. and B.-U. Choi in [3] uses similar parameter, but instead of purely using current weight, they use the *entropy* of the information can be extracted from the weight. These two approaches address the detection and recovery separately, as what we prefer.

One other method which addresses detection and recovery separately was proposed in our previous paper in [19]. This paper introduces a mixture method of MCW and two other intrinsic parameters; weight parameter and the standard deviation of particles.

This paper is a complement of the paper in which Landmark-based map is used to test the method. This paper test the method using a corridor-like map which confirms that the method is applicable in at least two types of map.

The rest of the paper is organized as follows. In chapter II Bayes filter and Monte Carlo Localization are briefly explained. Chapter III explains the definition of some terms used throughout the paper. Chapter IV reviews MCW method for kidnapping detection. The method we propose is then delivered in detail in Chapter V. Chapter VI delivers the simulations result of the comparison between the proposed method and MCW, and chapter VII gives the Conclusion.

II. BAYES FILTER AND MONTE CARLO LOCALIZATION

In mobile robot localization practice, it is almost impossible for a robot to know exactly its coordinates and heading (collectively known as *pose*) in the given map. Rather, the robot should infer them from the data from environment. The obtained state is then called *belief*. The belief of the robot is defined as

$$bel(S_t) = p(s_t | z_{1:t}, u_{1:t}) \quad (1)$$

This posterior is the probability distribution over the state $s_t = (x_t, y_t, \theta_t)$, at time t , given all past measurements $z_{1:t} = \{z_1, z_2, \dots, z_t\}$ and all past controls $u_{1:t} = \{u_1, u_2, \dots, u_t\}$. Sometimes it is also useful to consider the belief *before* the current measurement taken, that is

$$\overline{bel}(s_t) = p(s_t | z_{1:t-1}, u_{1:t}) \quad (2)$$

MCL is a Bayes-based localization algorithm. It provides a powerful tool to calculate posterior $bel(s_t)$, given measurement and control data [1], [11]. Bayes filter is based on Markov world assumption, i.e. past and future data are independent if one knows the current state s_t [1]. By implementing Bayes rule and this Markov world assumption, the belief posterior can be defined as

$$bel(s_t) = \eta p(z_t | s_t) \overline{bel}(s_t) \quad (3)$$

The term $p(s_t | s_{t-1}, u_{1:t})$ is defined as the *prediction* or *motion model*, since it reflects the state transition due to robot motion. The probability $p(z_t | s_t)$ itself is called *correction* or *sensor model*, since it incorporates sensor reading to update robot state. η is normalization constant, ensuring the final result to be normalized to one.

Bayes filter gives freedom to the choices of representation for the posterior. MCL represents the posterior $bel(s_t)$ by a set S_t of N weighted samples distributed according to the posterior [1], [3]. The density of the samples proportionally represents the likelihood of the robot's pose being there.

$$S_t = \langle s_t^{[n]}, \omega_t^{[n]} \rangle; \quad n = 1, 2, \dots, N \quad (4)$$

Each particle $s_t^{[n]}$ represents the hypothesis of the robot's pose at time t . The $\omega_t^{[n]}$ is the non-negative number called weight of particle. It indicates how good particle $s_t^{[n]}$ in representing the robot's pose.

The basic MCL algorithm, as summarized from [1] and [2] is depicted in table 1. It accepts previous state S_{t-1} , past controls u_t , past measurements z_t , and map information m .

Table 1. MCL Algorithm

| | |
|-----|---|
| 1. | MCL Algorithm (S_{t-1}, u_t, z_t, m) |
| 2. | $S_t = \overline{S}_t = \emptyset$ |
| 3. | for $n = 1$ to N do |
| 4. | generate $s_t^{[n]} \sim p(s_t s_{t-1}^{[n]}, u_{1:t}, m)$ |
| 5. | calculate weight $\omega_t^{[n]} = p(z_t s_t^{[n]}, m)$ |
| 6. | $\overline{S}_t = \overline{S}_t + \langle s_t^{[n]}, \omega_t^{[n]} \rangle$ |
| 7. | end for |
| 8. | normalize ω_t |
| 9. | for $n = 1$ to N do |
| 10. | draw $s_t^{[n]}$ with probability $\propto \omega_t^{[n]}$ |
| 11. | Add $s_t^{[n]}$ to S_t |
| 12. | end for |
| | return S_t |

III. DEFINITION OF TERMINOLOGIES

There are some terminologies throughout this paper which addresses kidnapping event. The first is the kidnapping detection. This process is defined as a process which detects the KRP event at any time t where

$1 < t \leq T$. We omit the first time step under common assumption that no kidnapping would occur at the very first time step of robot's operation. The detection process will only detect the time t of kidnapping, not the place the robot is being kidnapped to.

Kidnapping point is defined as the time instance t of when the robot is really kidnapped. The match between detected kidnapping and the kidnapping point will determine the success of KRP event detection. The successful detection has the following criterias:

1. The detection should occur only once, since we consider single kidnapping event only.
2. The time of detected kidnapping should be the same as the real kidnapping.

These two criteria are used to compare the accuracy between MCW and the proposed method. In this paper, the kidnapping itself is divided into two; Early Kidnapping and Late Kidnapping. The former one is the kidnapping which occurs during pre-convergence state, i.e. there are some particles spread around the map. The later one is the kidnapping in which the particles are already converged.

The last term is the recovery/re-localization process which is defined as a method to localize the robot after kidnapping event. We will use recovery and re-localization interchangeably in this paper.

IV. MAXIMUM CURRENT WEIGHT METHOD

The detection in MCW can be denoted by the following equation;

$$Kidnapped_t = \omega_t^{max} < \xi \quad (5)$$

Where ω_t^{max} is the maximum weight of the particle set at time t , while ξ is the threshold to detect the kidnapping.

This formulation leads to two problems;

A. Early Kidnapping Detection

If the robot is kidnapped early, the weight of particles will still be quite high. This is due to the fact that it is very likely there would be some particles at/near where the robot is being taken to (see Fig. 1). Therefore, the kidnapping around this time step will be very likely undetected if we rely on weight parameter. In Figure 1, the robot is being kidnapped at $t=3$. There is no recovery strategy applied, but we can see the robot can still re-localize itself as shown by the constant maximum weight of 1. This is again due to the existence of particles near the point where the robot is after being kidnapped.

As can be seen in Fig 1, the maximum weight at $t = 3$ is still high. This causes the inability of Eq. 5 to detect the kidnapping. Forcing this equation to detect the kidnapping at $t = 3$ by setting $\xi > \omega_3^{max}$ will make multiple detection outside $t = 3$, which means the detection fails as what had been discussed in chapter III. Kidnapping in this part (early kidnapping) does not affect the overall localization too much. However, sometimes an information of when the kidnapping happens can be used as fault detection such that

one will know something unnatural happens around the kidnapping point ($t=3$ in the example). MCW obviously fails at obtaining this information.

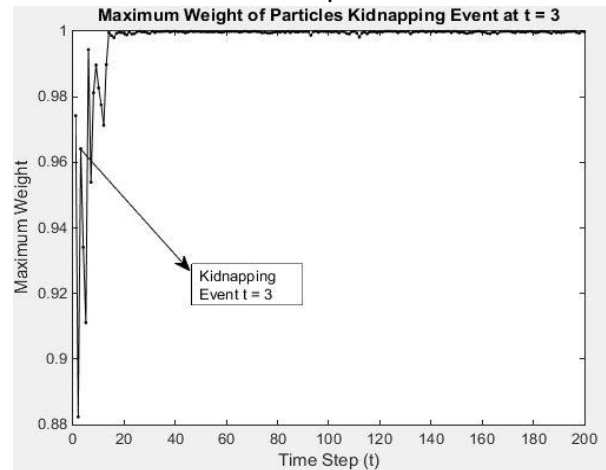


Figure 1. Maximum Weight of Particles with Kidnapping Event Happens at $t = 3$.

B. High Dependency to the Success of Recovery

There is also a high dependency between maximum current weight parameter and the ability of the recovery strategy. The weight of particle is based on how close the observation from particle to the observation from the robot. Therefore, when the recovery fails, the gap between two observations will be high and thus resulting in lower particles' weight (see Fig. 2).

From the figure we can see that when there is no recovery strategy (extreme case of recovery failure) to re-localize the robot, (the maximum weight drops very low. Forcing the detection to detect kidnapping will result in multiple detections at later time steps, thus the failure in detection.

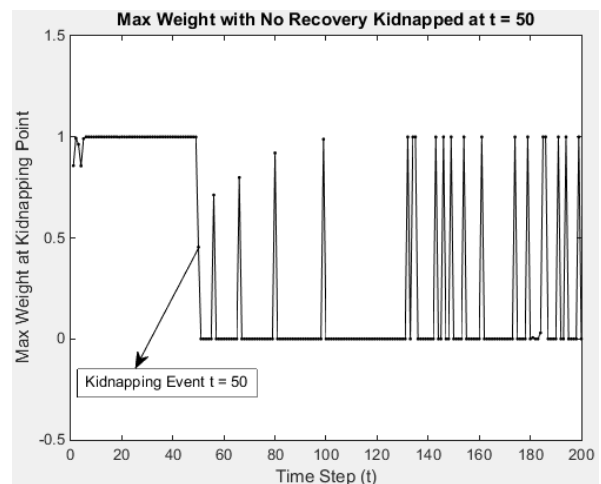


Figure 2. Max Weight of Particles with Kidnapping Event at $t = 50$ and no Recovery

The two drawbacks in MCW give motivation to devise a new detection strategy which can detect the kidnapping of the following nature:

- a) The detection should be able to detect kidnapping in large range of particle convergence

The detection should be highly independent to the success of the recovery, i.e. detection of kidnapping with high accuracy can still be obtained when the recovery fails.

V. PROPOSED METHOD

A new method of kidnapping event detection is proposed in [19] which uses three parameters as kidnapping detector and a particles' convergence detector to separate early kidnapping (EK) with late kidnapping (LK). The three kidnapping detectors are:

1. Change in standard deviation ($\Delta\bar{\sigma}_t$)
2. Change in mean of weight ($\Delta\bar{\omega}_t$)
3. Maximum Current weight (ω_t^{max})

The first parameter is used in early kidnapping case, and the latter two are for late kidnapping case. A convergence detector D_t is used to differentiate the two cases. This detector is defined as

$$D_t = \begin{cases} 1, & N_c > \varphi \\ 0, & elsewhere \end{cases} \quad (6)$$

Where N_c denotes the percentage of the number of particles exist within a unit distance from the mean. φ is the threshold to decide whether the particles is in convergence state or not. The complete kidnapping condition is then described as follows

$$Kid_{EK(t)} = \left(\neg \sum_1^t D_t \right) \wedge (\Delta\bar{\sigma}_t > \beta) \quad (7)$$

$$Kid_{LK(t)} = D_t \wedge (\omega_t^{max} < \xi) \wedge (\Delta\bar{\omega}_t < \alpha) \quad (8)$$

VI. SIMULATION RESULT

A series of simulations is designed to test the method. There are two types of map used in the test; the landmark-based map as presented in [19] and the corridor-like map. These two maps differ only in the way of weighting particles, how the sensor reading is used, and the limitation of robot's movement. The method is designed to not depend on these factor, thus the tests against these two maps should confirm it.

For each type of map, the proposed method is tested against the MCW to see which of the two is closer to the desired nature of kidnapping detection stated at the end of chapter IV. The tests are also divided into two categories to test the stated objectives. The first one is the kidnapping without recovery, and the the other one is kidnapping with recovery.

Each test is run to detect 200-time instances of kidnapping, $t_k = \{1, 2, 3, \dots, 200\}$. For each t_k , the simulation with 200-time steps is run for 100 times. The number of successful kidnapping detection is then calculated for each t_k , and divided by 100 to obtain the percentage of success rate.

The parameters of the detection method we test is presented in Table 2, while Fig 3 depict the corridor-like map in the simulations.

Table 2. Detection Method Parameters

| | Proposed Method | | | | MCW |
|-----------------|-----------------|-------|----------|---------|-------|
| | φ | ξ | α | β | ξ |
| Landmark | 70 | 0.1 | -0.3 | 0.05 | 0.1 |
| Corridor | 70 | 0.1 | -0.2 | 0.01 | 0.1 |

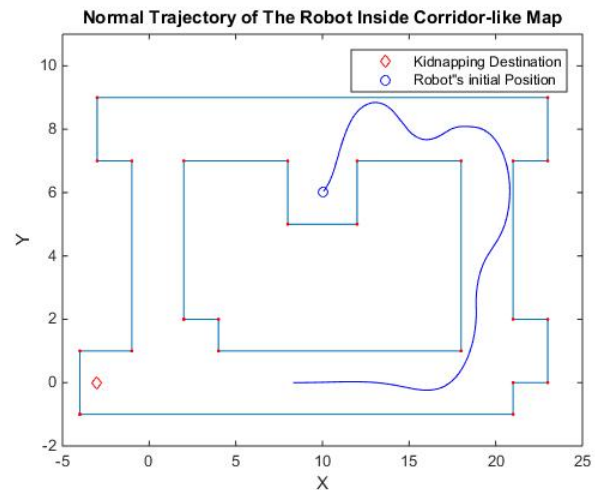


Figure 3. Corridor-like Map used in Simulations. The curvy line describes the robot's normal trajectory (Without Kidnapping)

A. Detection without Recovery

In order to test the independency of the methods towards recovery, a kidnapped robot problem is simulated without re-localization process. The test result is shown in Fig. 4.

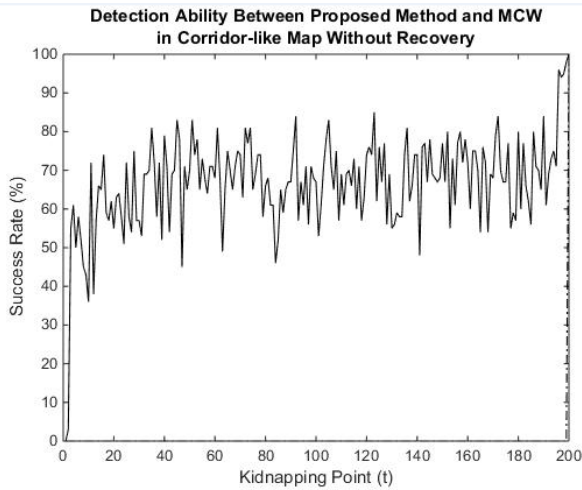


Figure 4. Success Rate per Kidnapping Point in Corridor-like map without recovery

In Fig. 4, it is proven that the detection accuracy of the proposed method does not depend heavily on the success of recovery while MCW clearly fails except at the very end of iteration ($t=200$).

B. Detection Test with Recovery

Because the MCW fails in detection without recovery, other type of tests are devised. In these tests, the recovery strategy is executed at the time step right after the kidnapping point ($t_{recovery} = t_k + 1$) regardless of the success of the detection of either method. The recovery strategy employed is the most basic one, which is the particles' re-initialization method. This recovery process is executed by replacing current set of particles by the randomly distributed particles drawn from uniform distribution over the pose space inside the map, the same as particles initialization at the very beginning of localization process. How the two methods in behave under this test is depicted in Fig. 7.

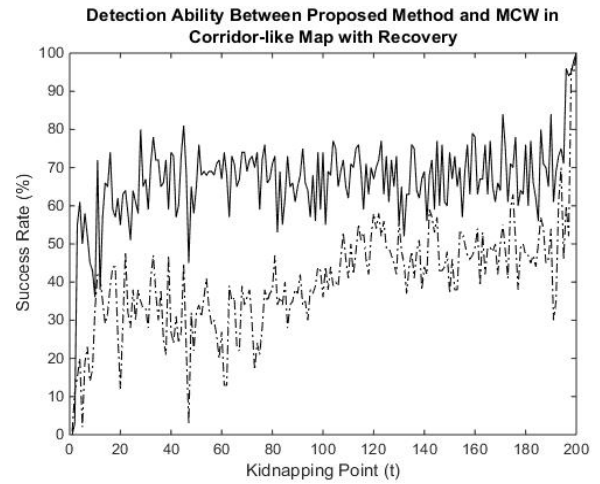


Figure 5. Success Rate per Kidnapping Point in Corridor-like Map with Recovery

The result in Fig.4 and Fig. 5 indicates that the proposed method still outperforms the MCW in corridor-like map. An interesting thing happens in $100 < t < 190$ for landmark-based map case. It can be seen that MCW even fails to correctly detect the kidnapping. In this area, the robot is kidnapped to the region outside the map (see Fig. 4 and eq. 9). Because the re-localization strategy is implemented only inside the map, the recovery most likely fails, thus a failure in detection by MCW.

VII. CONCLUSIONS

A new method in detecting the kidnapping event in Monte Carlo Localization is proposed. The method relies on three parameters, the change in weight of particles, the change in standard deviation of the particles, and the maximum current weight. A series of simulation tests comparison between the proposed method and pure Max Current Weight (MCW) is also conducted using corridor-like map. The results show that the proposed method can still detect the early kidnapping event with high accuracy when the MCW fails. This ability of kidnapping detection even when the recovery process itself is unneeded (characteristic of early kidnapping) shows that the proposed method is able to be a good fault detector independent to the needs of recovery.

The proposed method is also able to maintain high accuracy under the failure in recovery, showing that the independency towards the success of the recovery is very high compared to MCW. This independency is important to make sure that the information of when kidnapping happens is not disturbed by the success of recovery process.

These results complemented the results in [19] which shows the superiority against MCW in landmark-based map.

ACKNOWLEDGMENT

This work was supported in part by the Japan-ASEAN Integration Fund (JAIF) and AUN/SEED-Net under Grant No.R.J130000.7309.4B156 and Universiti Teknologi Malaysia under Grant no. Q.J130000.2509.05H54.

REFERENCES

[1] S. Thrun, W. Burgard and D. Fox, *Probabilistic Robotics* (MIT Press, Cambridge, MA, 2005).

[2] L. Zhang, R. Zapata and P. Lépinay, "Self-adaptive Monte Carlo localization for mobile robots using range finders," *Robotica* 30, pp.229–244 (2012).

[3] Yi, C. and B.-U. Choi, *Detection and Recovery for Kidnapped-Robot Problem Using Measurement Entropy*, in *Grid and Distributed Computing*, T.-h. Kim, et al., Editors. 2011, Springer Berlin Heidelberg, p. 293-299.

[4] Majdik, Andras, et al. "New approach in solving the kidnapped robot problem." *Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)*. VDE, 2010.

[5] Andreasson, Henrik, André Treptow, and Tom Duckett. "Localization for mobile robots using panoramic vision, local features and particle filter." *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. IEEE, 2005.

[6] Duan, Z.; Cai, Z.; Min, H. Robust Dead Reckoning System for Mobile Robots Based on Particle Filter and Raw Range Scan. *Sensors* 2014, 14, 16532–16562.

[7] T. Hester and P. Stone, "Negative Information and Line Observations for Monte-Carlo Localization," *Proceedings of IEEE International Conference on Robotics and Automation ICRA 2008*, Pasadena, CA (2008) pp. 2764-2769.

[8] S. Thrun, D. Fox, W. Burgard and F. Dellaert, "Robust Monte Carlo Localization for Mobile Robots," *Artif. Intell.* **128**(1-2), 99-141(2001).

[9] H. Andreasson, A. Treptow and T. Duckett, "Localization for Mobile Robots Using Panoramic Vision, Local Features and Particle Filter," *Proceeding of IEEE International Conference on Robotics and Automation (ICRA 05)*, Barcelona, Spain (2005) pp. 3348-3353.

[10] A. R. Vahdat, N. NourAshrafoddin and S. S. Ghidary, "Mobile Robot Global Localization Using Differential Evolution and Particle Swarm Optimization," *IEEE Congress on Evolutionary Computation (CEC 07)*, Singapore (2007) pp. 1527-1534.

[11] S. I. Roumeliotis and G. A. Bekey, "Bayesian Estimation and Kalman Filtering: A Unified Framework for Mobile Robot Localization," *Proceedings of IEEE International Conference on Robotics and Automation (ICRA '00)*, San Francisco, CA (2000) vol.3, pp. 2985-2992.

[12] S. Thrun, M. Beetz, M. Bennewitz, W. Buggard, A. Cremers, F. Dellaert, D. Fox, D. Hähnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz, "Probabilistic Algorithms and The Interactive Museum Tour-Guide Robot Minerva," *Int. J. Robot. Res.* **19**, 972-999(2000).

[13] L. Marchetti, G. Grisetti, and L. Iocchi, "A Comparative Analysis of Particle Filter Based Localization Methods," *RoboCup*, Bremen, Germany (2006), pp. 442-449.

[14] D. Fox, W. Burgard, H. Kruppa, and S. Thrun, "Efficient Multi-Robot Localization Based on Monte Carlo Approximation," In *Robotic Research: The Ninth International Symposium* (J. Hollerbach and D. Koditschek, Eds.) (Springer-Verlag, London, 2000).

[15] C. Kwok, D. Fox and M. Meila, "Real-Time Particle Filters," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 469-484, 2004.

[16] P. Pfaff, W. Burgard and D. Fox, "Robust Monte-Carlo Localization Using Adaptive Likelihood Models," *European Robotics Symposium*, Palermo, Italy (SpringerVerlag, 2006) pp. 181-194.

[17] C. Siagian and L. Itti, "Biologically-Inspired Robotics Vision Monte-Carlo Localization in the Outdoor Environment," *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems IROS 2007*, San Diego, CA, (2007) pp. 1723-1730.

[18] S. Thrun, D. Fox, and W. Burgard, "Monte Carlo localization with mixture proposal distribution," in *Proceedings of the AAAI National Conference on Artificial Intelligence*, 2000, pp. 859–865.

[19] I. Bukhori, Z.H. Ismail, and T. Namerikawa, "Detection Strategy for Kidnapped Robot Problem in Landmark-Based Map Monte Carlo Localization," in *IEEE International Symposium on Robotics and Intelligent Sensors* 2015.