

파티클 다양성 유지를 위한 지역적 그룹 기반 FastSLAM 알고리즘

Geographical Group-based FastSLAM Algorithm for Maintenance of the Diversity of Particles

장 준 영, 지 상 훈, 박 흥 성*

(June-Young Jang¹, Sang-Hoon Ji², and Hong Seong Park¹)

¹Dept. of Electronic and Communication Engr., Kangwon National University

²Dept. of Applied Robot Technology, Korea Institute of Industrial Technology

Abstract: A FastSLAM is an algorithm for SLAM (Simultaneous Localization and Mapping) using a Rao-Blackwellized particle filter and its performance is known to degenerate over time due to the loss of particle diversity, mainly caused by the particle depletion problem in the resampling phase. In this paper, the GeSPIR (Geographically Stratified Particle Information-based Resampling) technique is proposed to solve the particle depletion problem. The proposed algorithm consists of the following four steps : the first step involves the grouping of particles divided into K regions, the second obtaining the normal weight of each region, the third specifying the protected areas, and the fourth resampling using regional equalization weight. Simulations show that the proposed algorithm obtains lower RMS errors in both robot and feature positions than the conventional FastSLAM algorithm.

Keywords: FastSLAM, particle filter, particle's diversity, particle stratified, geographically grouped

I. 서론

SLAM (Simultaneous Localization and Mapping)은 로봇이 이동 중 알려지지 않은 환경에 대해 맵을 만들면서, 자동적으로 위치 추정을 하기 위한 로봇 자율 주행 분야에서 필수적인 기술이다[1-10,14,15].

이는 Localization과 비슷한 절차로 수행되나 발견된 위치와 자세정보를 확인할 수 있는 센서 특이정보인 특징점의 위치정보와 식별정보가 사전에 주어진 것이 아니라, 주행하면서 생성되고 추정된 정보라는 것이 다른 점이다. 그런데 로봇의 주행과정에서는 동작제어잡음과 환경센서 잡음이 존재하고 그 결과 예측된 로봇의 추정 오차가 커지게 되면서 SLAM의 효율이 매우 낮아지게 된다.

이러한 SLAM의 문제를 해결하기 위한 방법으로 칼만 필터를 사용하는 방법들이 연구되었는데, 특히 EKF-SLAM은 실내 환경[3-5], 실외 환경[6], 수중[7], 지하[8]등에서 로봇의 위치 인식에 많이 응용되었다. 그런데 이 방법에는 지도 확장에 따른 계산 복잡성의 증가 문제와 주행 프로세스 모델과 환경 센서 관측기 모델에서의 선형화 오차가 존재하고 이를 개선하기 위한 방안으로 IEFK-SLAM, UKF-

SLAM 등이 제시되었다. 그러나 데이터 연관에 대한 근본적인 문제점으로 인하여, 로봇 자세정보, 특히 헤딩 각도에 대한 오차가 크거나 주행 및 환경센서정보 특성이 수시로 변경되는 실제 환경에의 적용에는 많은 제약들이 존재한다. 이러한 문제들을 해결하기 위한 대안으로 FastSLAM이 제안되었다[10,14,15]. FastSLAM은 Rao Black-wellized Particle Filter를 사용하는 방법으로 파티클 마다 맵과 로봇의 위치로 분리하고, 파티클 필터를 사용하여 로봇의 위치를, 칼만 필터를 사용하여 특징점 위치를 추정한다. 그러므로 계산 복잡도가 특징점 수의 비례적으로 증가하며, 파티클 마다 다른 추정치를 저장함으로써, 추정치간의 보완이 가능한 장점이 있다.

하지만 FastSLAM은 시간이 지남에 따라 성능이 퇴보하는 단점이 존재한다. 이는 로봇의 위치를 추정하기 위한 파티클의 다양성을 상실하기 때문이다[11,12]. 즉 FastSLAM은 재추출 단계에서 낮은 가중치의 파티클들이 삭제되고 높은 가중치를 가진 파티클들을 삭제한 만큼 복제하게 되는데, 이 과정이 시간이 지남에 따라 공통된 파티클들이 많아지는 파티클 고갈 문제를 발생하게 하고, 이는 알고리즘의 일관성을 유지할 수 없게 만든다[13].

이와 같은 현상은 파티클의 수가 충분하지 않거나 획득된 환경 정보가 지나치게 많으면, 파티클의 다양성이 더 빨리 사라지게 되어 더욱 심각해진다. 또한 파티클의 다양성이 낮아지면 필터의 추정성능 저하로 연결되는 문제점이 있다. 그러므로 FastSLAM에서는 과도하게 재추출과정 일어나는 것을 방지하여 파티클의 다양성을 장기간 유지하기

* 책임저자(Corresponding Author)

Manuscript received June 20, 2013 / revised August 9, 2013 / accepted August 11, 2013

장준영, 박흥성: 강원대학교 전자통신공학과

(shiny2104@naver.com/hspark@kangwon.ac.kr)

지상훈: 한국생산기술연구원(robot91@kitech.re.kr)

※ 본 논문은 산업통상자원부 산업융합원천기술개발사업(과제번호: 10044006)에 의하여 지원된 연구 결과임.

위한 재추출 알고리즘이 필요하다.

위에서 언급한 문제와 같이 파티클의 다양성을 오랜 시간 동안 유지하기 위하여 다양한 연구들이 진행되어 왔다 [17-19]. 먼저 Gordon는 파티클의 수를 증폭하여 샘플의 수가 늘어남에 따라 정확한 샘플 수에 상응하는 확률 밀도 함수를 얻어 냈다[17]. 하지만 샘플수가 늘어나면 로봇의 예측에 대한 정확도가 상승하는 장점이 있지만, 재 추출 과정에서 발생하는 계산량이 증가하는 단점이 생긴다. 이에 따라서 Grisetti는 파티클 고갈 현상을 줄이기 위해 각 파티클과 제안 분포를 재사용함으로써 계산량을 줄이는 방법을 제안했다[18]. 하지만 로봇의 경로 및 환경 맵을 가중치가 가장 큰 파티클의 정보로만 나타내었고 이와 같은 도출 방법은 항상 일관성 있는 결과를 도출할 수 없었다. Xiuzhi는 재 추출 과정에서 제거되는 낮은 가중치를 가진 파티클의 그룹들을 높은 가중치를 가진 파티클의 그룹에 혼합하여 다양성 손실 문제를 개선하기 위하여 노력을 했다[19]. 하지만 다양한 파티클들을 유지할 수 있는 근본적인 해결책이 되지는 못했다.

본 연구에서는 재 추출 과정에서 발생하는 파티클의 다양성 상실 문제를 개선하기 위한 지역적으로 분산된 파티클 정보를 사용하는 GeSPIR (Geographically Stratified Particle Information-based Resampling) 기법을 제안한다. 이 방법은 파티클의 분포로부터 계산된 가중치와 파티클들이 위치한 지역적 정보를 이용하여 다양한 지역에 파티클들을 생존시켜 로봇위치 추정의 정확도를 향상시키며, 재 추출 과정의 영향력을 최소화 시켜 지속적으로 파티클의 다양성을 유지할 수가 있다.

본 논문의 구성은 다음과 같다. II 장에서는 FastSLAM의 구조와 다양성 상실 문제에 대해 기술한다. III 장에서는 지역적으로 분산된 파티클의 정보를 사용하는(GeSPIR) 알고리즘의 전체적인 내용에 대해 기술을 한다. IV 장에서는 컴퓨터 시뮬레이션을 통해, 기존 추출 알고리즘과 로봇의 위치 에러와 특징 점 에러 그리고 파티클의 수에 따른 위치 에러를 비교를 통해 GeSPIR의 향상된 성능을 입증한다. 마지막으로 V에서는 결론은 내린다.

II. 이동로봇의 FastSLAM 구조와 다양성 상실 문제

SLAM문제를 해결하기 위하여 FastSLAM에서는 시간 t 에서 주행로봇의 위치를 나타내는 s_t 와 맵 정보를 나타내는 Θ 를 기록한다. 맵 정보는 시간 t 에서 관찰된 n 개의 특징점들로 θ_n 으로 표시하고, 특징점의 수는 N_f 으로 표시한다. 기록된 정보를 통해 SLAM 문제는 확률 밀도 함수로 로봇의 위치와 지도의 위치를 Bayesian 법칙에 의해 $p(s_t, \Theta | z^t, u^t, n^t)$ 과 같이 공식화 될 수 있다. 여기에서 $p(s_t, \Theta | z^t, u^t, n^t)$ 은 맵 정보와 주행로봇의 위치의 posterior 밀도를 추정하기 위해 필요하며, 관측을 $z^t = \{z_1, \dots, z_t\}$, 로봇 제어 입력을 $u^t = \{u_1, \dots, u_t\}$, 그리고 맵의 위치 Θ 와 관측들 z^t 사이의 관계를 n^t 로 표현할 수 있다. 따라서 SLAM 문제는 Bayesian filtering 통해서 식 (1)로 표현될 수 있다 [10].

$$p(s_t, \Theta | z^t, u^t, n^t) = \int p(s_t | s_{t-1}, u_t) p(s_{t-1}, \Theta | z^{t-1}, u^t, n^t) ds_{t-1} \quad (1)$$

여기에서 FastSLAM은 로봇의 위치에 대한 Posterior가 알려졌을 때 효과적으로 식 (2)와 같이 인수분해 되어 질 수 있다[10,14-16].

$$p(s^t, \Theta | z^t, u^t, n^t) = p(s^t | z^t, u^t, n^t) \prod_{n=1}^{N_f} p(\theta_n | s^t, z^t, u^t, n^t) \quad (2)$$

환경 맵에서 각 특징점에 대해 FastSLAM은 파티클의 위치 $p(\theta | s^t, z^t, u^t, n^t)$ 에 대해 분리된 추정함수를 사용한다. 인수분해로 인하여, FastSLAM은 각 특징점마다 분리된 EKF를 유지할 수 있고, 따라서 EKF-SLAM보다 더 효율적으로 갱신된 추정 값을 얻을 수 있다. 시간 t 에서의 m 번째 파티클 정보 $s_t^{[m]}$ 은 식 (3)와 같이 나타낼 수 있다[14-16].

$$s_t^{[m]} = \langle s_t^{t,[m]}, u_{1,t}^{[m]}, \Sigma_{1,t}^{[m]}, \dots, u_{N_f,t}^{[m]}, \Sigma_{N_f,t}^{[m]} \rangle \quad (3)$$

여기에서 $[m]$ 은 파티클의 인덱스를 나타내고, $s_t^{[m]}$ 은 $[m]$ 번째 파티클의 경로 추정치, 그리고 $u_{N_f,t}^{[m]}, \Sigma_{N_f,t}^{[m]}$ 는 $s_t^{[m]}$ 번째 파티클과 관련된 특징점 위치를 나타내는 Gaussian의 평균과 공분산을 나타낸다.

FastSLAM 알고리즘 수행 과정은 추출(sampling), 측정데이터 갱신, 가중치부여, 재추출(resampling) 4단계를 반복적으로 수행한다. 추출 단계에서는 로봇이 이동 중 센서로부터 측정한 특징점에서 관찰한 파티클 $s_t^{[m]}$ 기록한다. 측정데이터 갱신 단계에서는 특징점들로부터 관찰한 파티클들이 새로운 파티클이 아니라면 각각의 관찰된 파티클에 대한 평균 $u_{N_f,t}^{[m]}$ 와 공분산 $\Sigma_{N_f,t}^{[m]}$ 갱신을 수행한다. 추출한 파티클 데이터를 갱신하고 나면, 각각의 파티클에 대한 가중치 $w_t^{[m]}$ 을 계산하며 식 (4)와 같이 나타낼 수 있다[10].

$$w_t^{[m]} = \frac{\text{Target Distribution}}{\text{Proposal Distribution}} \quad (4)$$

각각의 파티클에 가중치가 부여되면 가중치 $w_t^{[m]}$ 에 비례한 확률을 가진 파티클을 추출하고, 이를 대체하여 N 개의 파티클을 복제한다.

그림 1은 FastSLAM에서 일반적으로 사용하고 있는 반복적 중요도 재추출(SIR: Sequential Importance Resampling) 과정에 대해 보여준다[18]. 각각의 파티클은 대상 분포가 제안 분포 보다 큰 지역은 높은 가중치를 부여받고 작은 지역은 낮은 가중치를 부여 받는다. 그림 1에서 스텝이 $k-1$ 때 높은 가중치를 부여받은 임시적인 $[m+m]$ 번째 파티클 $s_t^{[m+m]}$ 은 재추출 과정에서 3배가 복제되고 낮은 가중치를 가진 $s_t^{[m]}$ 은 재추출 과정에서 자신과 동일한 개수가 복제되는 것을 확인 할 수 있다. 또한 일부 특징점들은 가중치가 낮게 부여되어 완전히 제거되는 것을 확인 할 수 있다. 이 과정이 반복됨에 따라 파티클들이 기록하고 있는 로봇 경로와 특징점의 추정 값을 반복적으로 상실하게 된다. 따라서 파티클 필터는 시간이 지남에 따라 독특한 특성을 가

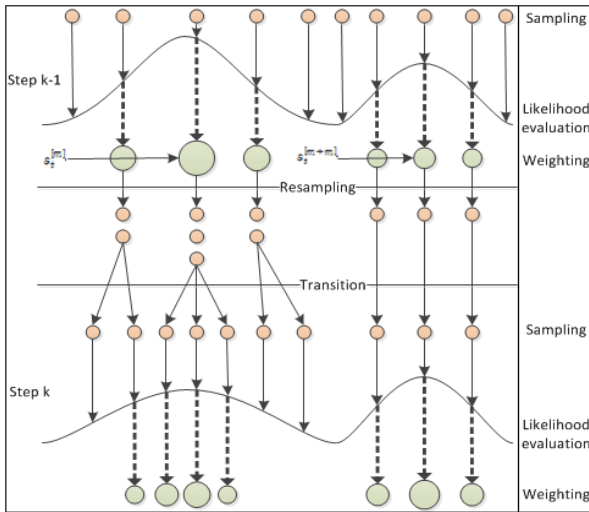


그림 1. FastSLAM에서 반복적 중요도 재 추출 과정.

Fig. 1. SIR process in FastSLAM.

진 파티클의 수가 줄어들게 되는 파티클 고갈 문제가 나타나는 것이다. 이것은 재추출 과정이 파티클의 다양성에 영향을 끼치는 중요한 과정임을 알게 해준다. 때문에 급격한 파티클의 다양성 손실은 필터의 일관된 장기 예측을 방지하게 되며, 추정 값의 정확도를 높이기 위해서는 재추출 때 제거되는 파티클의 영향을 최소화하는 방법이 필요하다.

III. 지역적으로 분산된 파티클의 정보를 사용하는 재 추출 기법 (GeSPIR)

본 논문에서는 FastSLAM에서 일반적으로 사용하는 SIR 알고리즘에 다양성 상실 문제를 개선하기 위하여 지역적으로 분산된 파티클의 정보를 사용하는 GeSPIR을 제안한다. 기존 알고리즘은 재 추출 과정이 반복됨에 따라 공통된 파티클이 많아지고 특정 지역에 다수의 파티클이 밀집하는 결과를 초래하여 필터의 추정 성능 저하를 유발한다. 그러므로 기존 재추출 알고리즘에서 발생할 수 있는 다양성 상실 문제를 개선하기 위하여 GeSPIR에서는 재 추출 과정을 수행하기 위하여 총 4단계의 과정을 거치게 된다.

그림 2는 GeSPIR의 4단계 과정에 대해 보여주고 있다. 먼저 파티클의 위치좌표를 이용하여 파티클 분포의 중심좌표를 계산한다. 이 좌표를 통해 K개의 지역으로 분할하여 각 지역에서 평균가중치를 가진 파티클을 재추출 과정에서 제외시킨다. 이 파티클들은 다음 로봇의 위치 예측 때 사용하기 위하여 복제된 후 저장하게 된다. 그리고 각 지역에 존재하는 파티클들의 위치좌표를 통해 K개의 중심좌표를 계산하여 각각의 중심 좌표 일정 범위 내에 존재하는 파티클들을 재 추출 과정에서 보호하기 위한 보호 지역을 설정한다. 이 보호 지역에 존재하는 파티클들은 재 추출 과정에서 제거되지 않으며 다양한 종류의 파티클들을 생존시킬 수 있다. 그리고 파티클의 분포를 통해 재추출 과정에서 낮은 분포지역에 존재하는 파티클에 지역적 평준화 가중치(regional equalization weight)를 부여하여 지역적으로 고르게 분포된 파티클 영역을 다음 로봇의 위치 예측 때에도 활용한다. 전체적인 GeSPIR 기법에 대한 방법은 알고리즘 1과 같다.

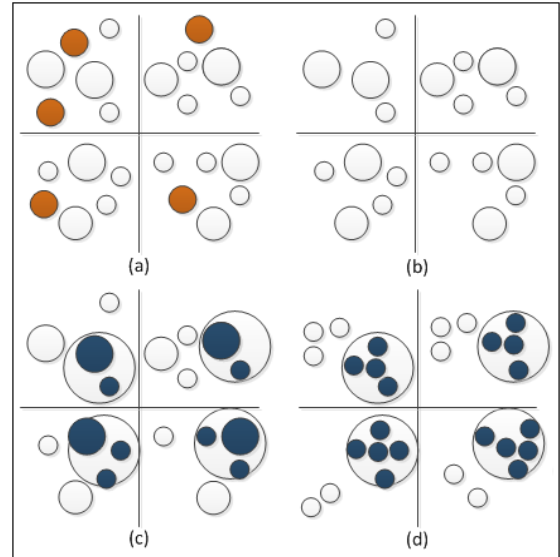


그림 2. GeSPIR 알고리즘 재 추출 과정. (a) 파티클 공간 분할 (b) 보통의 가중치를 가진 파티클을 재 추출 과정에서 제외 (c) 보호 지역 설정 (d) 지역적 평준화 수행.

Fig. 2. The GeSPIR algorithm resampling process. (a) Particle space split (b) Excluded from the Resampling process of normal weight of particle (c) Setting up of protected areas (d) Regional equalization is performed.

Algorithm 1. GeSPIR 알고리즘.

Algorithm 1. GeSPIR Algorithm.

Input	$*s_t^{[m]}$: particle information
output	$s_t^{[m]}$: new particle information
Steps	<ol style="list-style-type: none"> 1. Particles grouping [described in Fig. 2(a)] <ol style="list-style-type: none"> 1.1 using K-means algorithm [20,21] 1.2 calculate the coordinates of the center of the entire particle from $*s_t^{[m]}$ [Eq. (5)] 1.3 find the coordinates of the center of each region. [Eq. (6)] 1.4 record grouped K region each of particles $p(s^t, \Theta z^t, u^t, n^t, g^t)$ 1.5 the number of particles stored in the K region calculate $N_{[k]}$: From Each of particles recorded g^t 2. Find the middle weight from K region and save [described in Fig. 2(b)] <ol style="list-style-type: none"> 2.1 specify the weights of the threshold [Eq.(8)] 2.2 middle weight particles saving [Fig. 3] 3. Specify a range of safety [described in Fig. 2(c)] <ol style="list-style-type: none"> 3.1 setting up safety areas [Eq. (9)] 4. Resampling process [described in Fig.2(d)] <ol style="list-style-type: none"> 4.1 particle replication of safety areas 4.2 regional equalization weighted calculation [Eq.(12)] 4.3 input the new particles about n_{group} number
	$s_t^{[m]} \leftarrow *s_t^{[m]}$

다양한 지역에 고루 분포된 파티클을 일정 그룹으로 나누기 위하여 비 계층적 클러스터링 방법인 K-Means 알고리즘의 원리를 사용한다. 이는 클러스터의 수를 사전에 지정하고, 대상 객체들을 적절한 클러스터에 배정하는 것으로, n개의 객체가 주어졌을 때 주어진 객체가 지정한 클러스터 개수에 모두 포함할 때 까지 객체를 클러스터링하는 방법이다. 이 방법은 다양한 형태의 데이터에 적용이 가능하며 빠른 속도로 데이터를 클러스터링하는 장점이 있다[20,21].

먼저 파티클을 K개의 지역으로 클러스터링 하기 위하여 K개의 중심점을 설정할 필요가 있다. K개의 중심점은 전체 파티클의 분포의 중심 좌표로부터 나누어진 K개의 지역 안에 존재하는 파티클의 중심좌표로 설정한다. 먼저 K개의 지역으로 나누기 위해서 파티클의 중심좌표 α 를 알아야 한다. α 를 계산하기 위해서는 로봇이 이동 중 파티클의 위치를 측정하였을 경우 θ 의 범위에 따라 다르게 계산 되어야 한다. 이는 180도와 -180도를 평균으로 구하였을 경우 실제 평균은 180도가 되지만 값은 0도가 되기 때문이다. 그러므로 각도에 따라 좌표 값을 따로 저장해서 +영역의 값과 -영역의 값의 차를 더한 평균좌표를 파티클 분포의 중심좌표로 계산하였고 식 (5)와 같다.

$$\alpha = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \frac{\sum x_{t1} + \sum x_{t2}}{\sum w_{t1} + \sum w_{t2}} \\ \frac{\sum y_{t1} + \sum y_{t2}}{\sum w_{t1} + \sum w_{t2}} \end{bmatrix} \quad (5)$$

여기서 $\alpha[x,y]^T$ 는 전체 파티클 분포에서 중심좌표를, θ 가 0보다 크거나 같은 경우에 x 좌표 값을 x_{t1} , y 좌표 값을 y_{t1} , 가중치를 w_{t1} 라 한다. 그리고 θ 가 0보다 작은 경우에 x 좌표 값을 x_{t2} , y 좌표 값을 y_{t2} , 가중치를 w_{t2} 라 한다. 파티클의 좌표 값의 합과 가중치의 합의 평균을 이용하여 파티클 분포의 중심좌표로 설정하면 K개의 클러스터를 구성하기 위하여 K개의 중심점 $center(x,y)_k$ 을 설정하고 그 중심점과의 거리 d_k 를 계산하게 되며 이는 식 (6)과 같다.

$$\begin{aligned} center(x,y)_k &= mean(\sum_{d \in k} \beta) \\ d_k &= d(\alpha, center(x,y)_k) \end{aligned} \quad (6)$$

여기서 α 는 전체 파티클 분포의 중심 좌표, β 는 K지역에 속해 있는 파티클의 좌표이다. d_k 는 α 와 $center(x,y)_k$ 의 거리 값이 된다. 여기서 파티클의 거리를 측정하는 방법에는 일반적으로 사용되고 있는 Euclidean distance를 사용 하며 식 (7)과 같다.

$$d(x,y)_k = \left[\sqrt{\sum_{i=1}^{N_p} (p_i - q_i)^2} \right]_K \quad (7)$$

각 지역으로부터 중심 좌표가 정해지면 중심 좌표로부터 가까운 거리에 위치한 파티클들을 그룹화하고 파티클들은 자신이 속해있는 그룹에 대한 그룹번호를 부여 받게 된다. 파티클의 그룹화가 끝나면 각각의 그룹에서 존재하는 평균적인 가중치를 가진 파티클을 찾는다. 이는 파티클이 가지

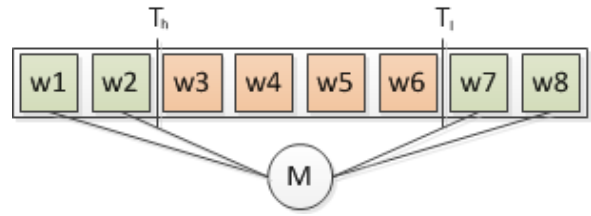


그림 3. 평균적인 가중치를 보존하기 위한 Threshold 값 지정.
Fig. 3. Specify the value of the threshold in order to preserve the weighted average.

고 있는 가중치 정보를 통해 내림차순으로 파티클을 정렬하고 높은 가중치의 판단 기준인 T_h 와 낮은 가중치의 판단 기준인 T_l 의 사이 값을 가지는 파티클을 재 추출 과정에서 제외시킨다. 각각을 구하는 방법은 식 (8)과 같다.

$$T = \begin{bmatrix} T_h \\ T_l \end{bmatrix} = \begin{bmatrix} T_{size} - (T_{size} * S_{late}) \\ T_{size} * S_{late} \end{bmatrix} \quad (8)$$

여기서 T_{size} 는 파티클의 개수, S_{late} 가중치의 상위 그룹과 하위 그룹으로 나누기 위한 공간을 할당하는 사용자 입력 값인 0.1~0.9의 값을 가지는 수치이다. 평균적인 가중치를 가지는 파티클은 복제과정을 거친 후 $s_i^{[m]}$ 에 저장되어 다음 스텝 때 계속적으로 로봇의 위치 예측에 활용될 수 있으며 T 를 통해 평균적인 가중치를 획득하는 방법과 나머지 파티클 그룹인 M 을 획득하는 방법은 그림 3과 같다.

평균적인 가중치를 가지는 파티클을 제외하고 남은 파티클 중에서 식 (6)과 식 (7)로 부여 받은 그룹 번호로부터 그 그룹에 속해있는 파티클의 분포를 기록하기 위하여 $N_{[k]}$ 를 계산하게 되는데, 이는 파티클의 지역적 분포를 평균화하기 위한 가중치를 적용할 때 이용된다. 또한 식 (6)으로부터 계산된 $centroid(x,y)_k$ 와 d_k 로 보호 지역을 설정하게 된다. 이 보호지역에 존재하는 파티클은 제거될 수 있는 파티클이라도 유지되어 파티클을 다양성을 유지시킨다. 보호지역에 존재하는 파티클을 판단하는 조건은 식 (9)와 같다.

$$d(center(x,y)_k, \beta) < d_k * safetyRadius \quad (9)$$

즉 보호 지역은 K지역 안에 있는 파티클의 분포 중심 좌표인 $center(x,y)_k$ 와 K지역에 속하는 파티클의 좌표인 β 의 거리가, 전체 파티클 분포의 중심 값에서 K지역에 존재하는 파티클의 중심 좌표와의 거리인 d_k 에 보호지역 설정 값인 $safetyRadius$ 를 곱한 값 보다 작은 거리에 존재하는 파티클에 대해 보호 지역에 존재하고 있는 파티클로 판단하게 된다. 일반적으로 본 논문에서는 상위 95%에 해당하는 파티클을 보호 지역에 존재하는 파티클로 판단하였다. 보호 지역이 설정되면 각 파티클에 대한 복제 개수를 설정하게 된다. 복제개수 $n_{Partides}$ 을 정하는 과정은 식 (10)과 같다.

$$n_{Partides} = \left[M \frac{w^{[p]}}{\sum_{p=1}^M w^{[p]}} \right] \quad (10)$$

만약 식 (10)에 의해 보호지역에 존재하는 파티클에 대해서 파티클의 복제 개수가 1보다 작고 파티클이 생존하지 못할 때 복제 개수 $n_{SafeZone}$ 을 부여하게 되고 이는 식 (11) 과 같다.

$$n_{safezone} = \begin{cases} n_{partides} & , n_{partides} \geq 1 \\ n_{partides} + 1 & , otherwise \end{cases} \quad (11)$$

마지막으로 보호지역에 설정된 파티클을 제외한 남은 파티클의 복제 개수를 통해서 K개 지역에 분포되어 있는 파티클의 개수를 일정 수로 유지시켜 지역적으로 평균화하기 위한 가중치 n_{group} 을 구하게 되고 이는 식 (12)와 같다.

$$n_{group} = \left\lceil n_{partides} * \left(\frac{M/K}{M} - \frac{N_{[k]}}{M} + 1 \right) \right\rceil \quad (12)$$

여기서 식 (5)와 식 (6)을 통해 구한 그룹화 과정으로 얻은 K개의 지역에 속해있는 파티클의 수인 $N_{[k]}$ 와 파티클을 그룹화 한 개수인 K를 이용하여 파티클의 수가 적은 지역에 위치한 파티클에 대해서 좀 더 높은 복제 개수를 부여한다. 여기서 $(M/K)/M$ 은 평균가중치를 제외한 파티클의 개수에서 각 지역에 분포해야할 파티클의 평균 수치를 구하는 식이다. 즉 100개의 파티클을 4개의 지역으로 나누었을 경우 각 지역마다 존재해야할 파티클의 일정 수는 $(100/4)/4$ 로 0.25가 된다. 그리고 $N_{[k]}/M$ 은 각 지역에 존재하는 파티클의 실제 수를 구하기 위함이다. 따라서 식 (12)로부터 실제 파티클의 분포도를 이용하여 가산가중치를 부여하게 되고 이는 최종적으로 파티클이 복제되어야할 개수에 이용되어 평균적인 가중치를 가진 파티클과 보호지역에 설정된 파티클 그리고 지역적평준화 과정을 수행 후 남은 새로운 파티클 집합 $s_i^{[m]}$ 을 구성하게 된다. 때문에 지역적으로 파티클의 분포가 낮고 가중치가 낮은 경우에도 가산가중치가 부여되어 재 추출 과정에서 생존할 수가 있다.

IV. 실험 및 결과

GeSPIR의 성능 평가를 하기 위해 CPU 3.30GHZ에 RAM 4.0GB를 가지는 PC에서 시뮬레이션을 구성하였고 그림 4와 같다. 그림 4는 12개의 특징점을 가진 지도에 있어서 주어진 경로를 실선으로 표현하고, 제안한 알고리즘을 활용하여 예측된 경로를 점선으로 표현하여, 1회 폐 루프가 일어났을 때 예측된 로봇의 경로를 표현한 예이다. 로봇 바퀴의 반지름은 0.2m이고 바퀴와 바퀴사이의 거리는 0.4m로 설정하였다. 로봇의 이동속도는 일정한 각도로 회전하기위하여 왼쪽이 3.5m/s 오른쪽이 4m/s로 설정하였고, 바퀴의 대한 오도메트리 노이즈는 0.02m/s로 설정하였다. 각각의 로봇이 이동하였을 경우에 대한 특징점의 오차와 로봇 위치에 대한 오차를 기록하였고 25개 50개 75개 100개의 파티클들을 이용하여 총 10번을 반복적으로 측정하여 오차에 대한 평균을 기록하였다. 로봇이 이동할 때, 식 (3)에 의해 $s_i^{[m]}$ 의 모든 파티클에 대하여 노이즈를 포함한 로봇의 상태는 식 (13)과 식 (14) 그리고 식 (15)으로 표현할 수 있다.

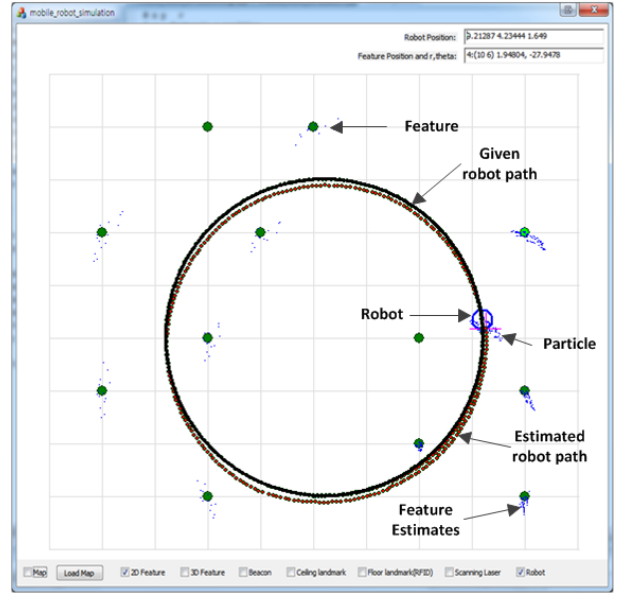


그림 4. 시뮬레이션 환경에서 주어진 경로 및 예측된 경로.
Fig. 4. Given path and estimated path in Simulation environment.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \Delta s^{[m]} \cos(\theta^{[m]} + \Delta\theta^{[m]}/2) \\ \Delta s^{[m]} \sin(\theta^{[m]} + \Delta\theta^{[m]}/2) \\ \Delta\theta^{[m]} \end{bmatrix} \quad (13)$$

$$\Delta s^{[m]} = \frac{\Delta s_r^{[m]} + \Delta s_l^{[m]}}{2}, \Delta\theta^{[m]} = \frac{\Delta s_r^{[m]} - \Delta s_l^{[m]}}{b} \quad (14)$$

로봇의 상태는 (x, y, θ) 로 모델링 될 수 있으며 여기서 (x, y) 는 직교 좌표 θ 는 방향이다.

$$\begin{aligned} \Delta s_r^{[m]} &\leftarrow \text{gauss rand}(\Delta s_r, k_r | \Delta s_r) \\ \Delta s_l^{[m]} &\leftarrow \text{gauss rand}(\Delta s_l, k_l | \Delta s_l) \end{aligned} \quad (15)$$

노이즈를 고려한 좌우 바퀴의 이동양을 식 (16)과 같이 표현될 수 있다. 여기서 $\Delta s_r, \Delta s_l$ 은 마지막 추출 기간 동안 각각 오른쪽 바퀴와 왼쪽 바퀴가 회전한 양이고, k_r, k_l 은 좌우 바퀴의 에러상수이며 0.02/ms 설정하였다. 평균 u 에서 분산 Σ 를 가지는 가우시안 랜덤 값은 $\text{gauss rand}(u, \Sigma)$ 이다. 로봇의 위치에서 특징점을 측정하였을 경우의 센서의 측정 모델 r_i 와 θ_i 는 식 (16)과 같다.

$$\begin{bmatrix} r_i \\ \theta_i \end{bmatrix} = \begin{bmatrix} \sqrt{(x-x_i)^2 + (y-y_i)^2} + w_r \\ \tan^{-1} \frac{y-y_i}{x-x_i} - \theta + w_\theta \end{bmatrix} \quad (16)$$

여기에서 (x_i, y_i) 는 맵에 대한 특징점의 위치이며, (x, y) 로봇이 이동 중 특징점을 관찰 후 예측한 파티클의 위치 $W = [w_r, w_\theta]^T$ 는 관찰 노이즈이다. 따라서 r_i 와 θ_i 는 특징점에 대한 관찰 노이즈를 고려한 거리와 방향이 된다. SLAM은 로봇의 위치인식과 맵의 올바른 작성을 요구하는 문제이기 때문에 위치 오차에 대한 에러를 측정이 매우 중요하다[22]. 그러므로 각각의 위치 오차에 대한 에러는 식 (17)과 식 (18)를 통해서 계산하였다.

$$F = \sqrt{\frac{1}{N_p \cdot N_f} \sum_{m=1}^{N_p} \sum_{i=1}^{N_f} (\theta_i - u_{i,t}^{[m]})^2} \quad (17)$$

$$P = \sqrt{\frac{1}{N_p} \sum_{m=1}^{N_p} (x_t^{true} - x_t^{[m]})^2} \quad (18)$$

여기에서 F 와 P 는 각각 RMS (Root Mean Square) 특징 점에러와 RMS 위치 에러를 나타낸다. θ_i 와 x_i^{true} 는 특징 점 i 와 로봇의 실제 위치이고, $x_t^{[m]}$ 은 m 번째 파티클에서 추정된 로봇의 위치 그리고 N_p 는 파티클의 수이다. 여기서 구해진 F 와 P 를 이용하여 측정된 누적값의 평균을 구하여 그래프로 표현하였다. GeSPIR 알고리즘은 기존 FastSLAM에서 일반적으로 사용하는 SIR 알고리즘과 PSR (Partial stratified Resampling)을 이용하여 재 추출 과정에서 살아남은 파티클의 수의 비교, 위치 오차(RMS position error), 특징점 오차(RMS feature error) 그리고 파티클의 수에 따른 위치 에러를 비교함으로써 증명하였다. 각각의 그래프는 GeSPIR이 실선, PSR이 빈 점선, 그리고 SIR이 점선이다.

그림 5는 초기 생성된 파티클에 인덱스를 부여하여 재 추출 과정에서 살아남은 파티클의 종류를 측정한 그래프이다. 재 추출 과정이 발생하였을 경우 GeSPIR의 경우는 초기 60개의 파티클을 유지하면서 로봇의 위치를 업데이트 하는 것을 볼 수가 있다. 재 추출과정이 반복적으로 진행됨에 따라 파티클의 다양성은 감소하지만 평균 30개의 파티클을 유지하면서 로봇의 위치를 예측하고 있었다. 반면 기존 알고리즘의 경우는 급격하게 파티클의 종류가 감소하는 것을 그래프를 통해 확인 할 수 있었다.

그림 6은 GeSPIR과 기존 재추출 알고리즘과 로봇 위치 에러의 누적값 평균의 오차를 비교한 그래프이다. 시간이 약 0~200까지 모든 알고리즘은 RMS 위치 에러가 급격하게 증가하는 것을 확인할 수 있다. 이는 초기 로봇이 SLAM 알고리즘을 수행하는데 있어서 많은 특징점이 보여지고 새롭게 파티클을 생성하기 때문에 생기는 현상이다. 로봇의 Close Loop가 일어났을 경우 생성된 파티클들은 갱신 과정에서 위치적으로 보정을 수행한다. Close Loop가 일어나기 전까지 SIR은 약 0.23m, PSR은 약 0.20m, GeSPIR은 약 0.15m를 보였다. 시간이 200 이후에는 모든 알고리즘은 오차를 보정 하고 있음을 확인 할 수 있다. 그래프에서 보는 바와 같이 제안한 알고리즘인 GeSPIR이 200~600 사이에 최솟값 0.1m를 보인 반면에 SIR과 PSR은 각각 0.20m, 0.16m를 보였다. 600시간 이후부터는 GeSPIR은 0.07m로 SIR이 0.18m로 수렴하고 있음을 볼 수 있었으며 PSR은 0.14m로 수렴하고 있음을 확인 할 수가 있다. 그래프에서 확인 할 수 있는 것과 같이, GeSPIR 알고리즘은 다른 알고리즘에 비해 낮은 위치 오차를 유지하면서 안정적으로 로봇의 위치를 보정하고 있음을 확인 할 수 있다.

그림 7은 GeSPIR, SIR, 그리고 PSR의 RMS 특징점 에러의 누적 값의 평균을 보인 그래프이다. 모든 알고리즘은 약 0~150까지 특징점에 대한 어려움이 증가하고 있다. 이 지역

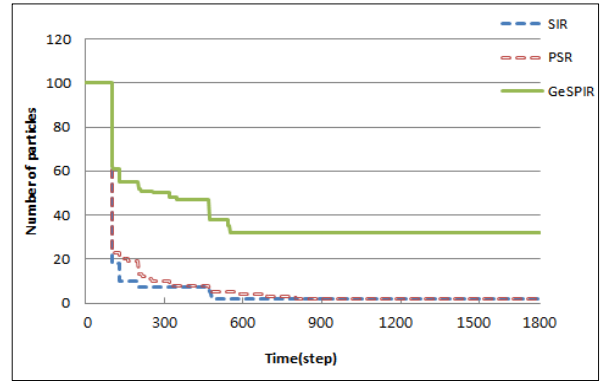


그림 5. 재 추출 과정에서 살아남은 파티클의 수 측정.
Fig. 5. Kind of measure of the surviving particles in the resampling process.

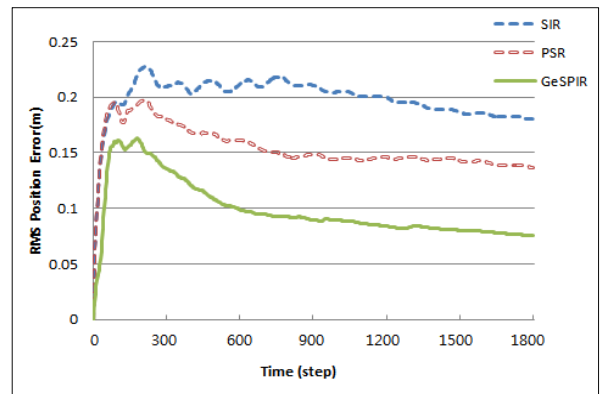


그림 6. RMS 위치 오차 비교.
Fig. 6. RMS position error comparison.

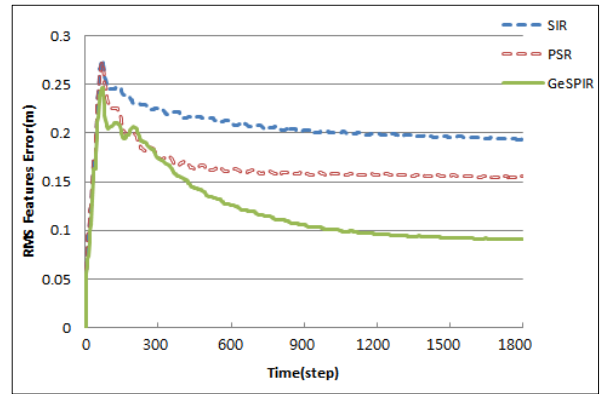


그림 7. 재추출 알고리즘에 대한 RMS 특징점 오차 비교.
Fig. 7. Comparison between the RMS features error of resampling algorithms.

에서 SIR과 PSR은 각각 0.28m와 0.27m를 보였다. 반면 GeSPIR은 0.25m로 가장 낮은 수치를 보였다. 150~600 사이에는 SIR이 최솟값이 약 0.22m를 보였으며 GeSPIR이 PSR보다 약 270 시간 때 상승 하였지만 시간이 지날수록 각각 최솟값 GeSPIR이 0.13m, PSR이 0.16m로 GeSPIR이 빠르게 위치를 보정하고 있음을 확인 할 수 있었다. 600 스텝 이후

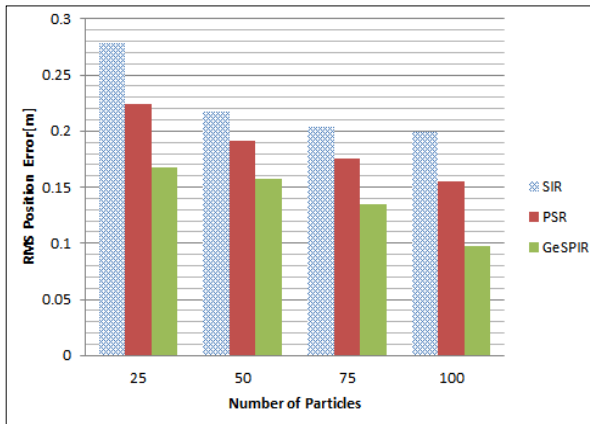


그림 8. 파티클 수에 따른 평균 RMS 위치 오차.

Fig. 8. Average RMS error according to the number of particles.

에는 SIR이 0.2m로 수렴하고 있었고 PSR은 0.15m GeSPIR은 0.1m로 수렴하고 있었다. 이는 높은 가중치를 가진 파티클을 반복적으로 복제 할 경우 실제 로봇 위치를 추정하는데 있어서 오차를 반영할 수 있다는 것을 확일 할 수가 있다. 따라서 다른 두 재추출 알고리즘에 비하여 낮은 위치 오차를 유지하면서 안정적으로 로봇의 위치를 보정하고 있음을 확인 할 수 있다.

그림 8은 파티클 수에 따른 RMS 위치 오차를 기록한 표이다. 파티클의 수는 로봇의 위치를 예측하는데 있어 다수의 샘플이 존재할 경우 좀 더 정확한 위치를 예측할 수 있다. 하지만 파티클의 수가 많아지면 계산량이 증가하여 시스템의 성능이 저하될 수 있는 단점이 있다. 파티클 개수에 대한 샘플 개수를 각각 25, 50, 75, 100를 유지하였을 때 최댓값 0.293m 최솟값 0.199m를 보인 SIR과 최댓값 0.224m 최솟값 0.155m를 보인 PSR에 비해 GeSPIR에서는 최댓값 0.168m 최솟값 0.098m를 나타내고 있다. 이는 파티클의 수치가 적음에도 불구하고 기존 알고리즘에 비해 로봇의 RMS 위치 에러를 효과적으로 줄이고 있음을 알 수 있다.

V. 결론

기존 SLAM의 주요한 방법이었던 EKF-SLAM에 비하여 RBPF (Rao-Blackwillized Particle Filter)를 사용하는 FastSLAM은 강력한 추정 성능의 장점을 가진다. 그러나 이러한 FastSLAM도 시간이 지남에 따라 성능이 저하되는 단점을 가진다. 성능 저하의 주요한 원인은 재추출 과정이 반복적으로 진행됨에 따라 파티클이 다양성을 상실함으로써 나타나는 파티클 고갈 문제 때문이다. 이 현상은 파티클의 수가 충분하지 않거나 획득된 환경 정보가 지나치게 많으면, 더욱 심각해진다. 또한 파티클의 다양성이 낮아지면 필터의 추정성능 저하로 연결되는 문제점이 있다.

본 논문에서는 파티클 고갈 문제를 개선시키고 파티클의 다양성을 확보하기 위하여 지역적 그룹 기반 재추출 기법인 GeSPIR를 제안하였다. 이 기법은 파티클의 분포로부터 계산된 가중치와 파티클들이 위치한 지역적 정보를 이용하여 다양한 지역에 파티클들을 생존시켜 로봇위치 추정의 정확도를 향상시키며 재추출 과정의 영향력을 최소화 시

켜 지속적으로 파티클의 다양성을 유지하기 위함이었다. GeSPIR 알고리즘의 성능은 시뮬레이션을 통하여, 기존 재추출 기법이었던 SIR과 PSR의 RMS position error와 feature error, 파티클의 수에 따른 로봇 위치 오차 그리고 재추출 과정에서 살아남은 파티클의 종류를 비교함으로써 알고리즘의 우수성을 보였다.

REFERENCES

- [1] W. Lu and C. Zixing "Progress of CML for mobile robots in unknown environments," *Robot*, vol. 26, no. 4, pp. 380-384, Jul. 2004.
- [2] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part1," *IEEE Robotics and Automation Magazine*, vol. 13, no. 2, pp. 99-110, Jun. 2006.
- [3] J.-T. Park and J.-B. Song "Low-cost sensor-based exploration in home environments with salient VVisual features," *International Conference on Control, Automation and Systems*, pp. 2218-2222, Oct. 2010.
- [4] S.-Y. An and J.-G. Kang, "SLAM with visually salient line features in indoor hallway environments," *Journal of Institute of Control, Robotics and System (in Korean)*, vol. 16, no. 1, pp. 40-47, Jan. 2010.
- [5] S.-Y. Hwang and J.-B. Song "Monocular vision and odometry-based SLAM using position and orientation of ceiling lamps," *Journal of Institute of Control, Robotics and Systems (in Korean)*, vol. 17, no. 2, pp. 164-170, Feb. 2011.
- [6] T. Bailey, "Mobile robot localization and mapping in extensive outdoor environments," Ph.D. Thesis, University of Sydney, NSW, Australia, 2002.
- [7] S. Williams, G. Dissanayake, and H. F. Durrant-Whyte, "Towards terrain-aided navigation for underwater robotics," *Advanced Robotics*, vol. 15, pp. 533-549, Apr. 2012.
- [8] S. Thrun, D. Hahnel, D. Ferguson, Montemerlo, and R. Triebel, "A system for volumetric robotic mapping of abandoned mines," *Proc. of IEEE International Conference on Robotics and Automation*, Taipei, pp. 4270-4275, Sep. 2003.
- [9] S. Thrun, W. Burgard, and D. Fox, Probabilistic, Cambridge, MIT Press, 2005.
- [10] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM 2.0 : an improved particle filtering algorithm for simultaneous localization and mapping that probably converges," *Proc. of International Joint Conference on Artificial Intelligence*, 2003.
- [11] M. Bolic, P. M. Djuric, and S. Hong, "Resampling algorithms for particle filters: a computational complexity perspective," *Journal on Applied Signal Processing*, pp. 2267-2277, Apr. 2004.

- [12] L. Zhang, X. Meng, and Y. Chen, "Convergence and consistency analysis for FastSLAM," *IEEE Intelligent Vehicles Symposium*, pp. 447-452, June. 2009.
- [13] T. Bailey, J. Nieto, and E. Nebot, "Consistency of the FastSLAM algorithm," *IEEE International Conference on Robotics and Automation*, pp. 424-429, May 2006.
- [14] S. Thrun, M. Montemerlo, D. Koller, B. Wegbreit, J. Nieto, and E. Nebot, "FastSLAM: An efficient solution to the simultaneous localization and mapping problem with unknown data association," *Journal of Machine Learning Research*, vol. 4, pp. 380-407, 2004.
- [15] M. Montemerlo, "FastSLAM: a factored solution to the simultaneous localization and mapping problem with unknown data association," Ph.D. Thesis, Carnegie Mellon University, pp. 593-598, 2003.
- [16] R. Havangi and M. A. Nekoui, "An improved FastSLAM framework using soft computing," *Turk J Elec Eng & Comp Sci*, vol. 20, pp. 25-46, Apr. 2012.
- [17] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-gaussian Bayesian state estimation," *IEEE-Proceeding-F*, vol. 140, pp. 107-113, Oct. 1993.
- [18] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *Proc. of IEEE Trans. Robot*, vol. 23, no. 1, pp. 34-46, Feb. 2007.
- [19] X. Li, S. Jia, and W. Cui, "On sample diversity in particle filter based robot SLAM," *Proc. of the 2011 IEEE Internatinal Conference on Robotics and Biomimetics*, Phuket pp. 1072- 1077, Dec. 2011.
- [20] D. Napoleon and P. G. Lakshmi "An efficient K-means clustering algorithm for reducing time complexity using uniform distribution data points," *IEEE Trendz in Information Sciences & Computing (TISC)*, pp. 42-45, Dec. 2010.
- [21] M.-U.-S. Shameem, "An Efficient K-Means algorithm integrated with Jaccard distanc measure for document clustering," *IEEE Conference on AH-ICI*, pp. 1-6, Nov. 2009.
- [22] N. Kwak, G.-W. Kim, and B.-H. Lee, "A new compensation technique based on analysis of resampling process in FastSLAM," *Robotica*, vol. 26, no. 2, pp. 205-257, Mar. 2008.

**장 준 영**

2011년 강원대학교 전자통신공학과 졸업. 2011년~현재 동 대학원 석사과정. 관심분야는 로봇 자율 주행, 위치 인식, 로봇 소프트웨어 플랫폼, 로봇 시뮬레이션, 로봇제어 스크립트 언어.

**지 상 훈**

1995년 서울대학교 제어계측공학과 졸업. 1997년 동 대학원 전기공학부 석사. 2007년 동 대학원 전기컴퓨터공학과 박사. 2008년~현재 한국생산기술연구원 2010년~현재 대한전기학회 정보제어 부문 이사.

**박 흥 성**

1983년 서울대학교 제어계측공학과 학사 졸업. 1986년 동 대학원 석사. 1992년 동 대학원 박사. 1992년~현재 강원대학교 전자통신공학과 교수. 관심분야는 로봇 S/W 플랫폼, 무선데이터통신, 실시간 통신.