

실내 복도 환경에서 선분 특징점을 이용한 비전 기반의 지도 작성 및 위치 인식

SLAM with Visually Salient Line Features in Indoor Hallway Environments

안수용, 강정관, 이래경, 오세영*

(Su-Yong An, Jeong-Gwan Kang, Lae-Kyeong Lee, and Se-Young Oh)

Abstract: This paper presents a simultaneous localization and mapping (SLAM) of an indoor hallway environment using Rao-Blackwellized particle filter (RBPF) along with a line segment as a landmark. Based on the fact that fluent line features can be extracted around the ceiling and side walls of hallway using vision sensor, a horizontal line segment is extracted from an edge image using Hough transform and is also tracked continuously by an optical flow method. A successive observation of a line segment gives initial state of the line in 3D space. For data association, registered feature and observed feature are matched in image space through a degree of overlap, an orientation of line, and a distance between two lines. Experiments show that a compact environmental map can be constructed with small number of horizontal line features in real-time.

Keywords: RBPF, SLAM, line features, mobile robot

I. 서론

이동 로봇의 동시 위치인식 및 지도작성 (SLAM: Simultaneous Localization and Mapping)은 로봇이 미지의 영역을 주행하면서 주변 환경을 인지하고 그 환경에 대한 로봇의 상대적인 위치를 동시에 추정하는 것이다[1]. 지난 20년간 SLAM 알고리즘에 대한 여러 가지 연구들이 수행되어 왔으며, SLAM 알고리즘은 로봇의 위치와 지도의 상태(state)를 예측하는 방법으로 나누어 볼 수 있다.

로봇과 지도의 상태 예측 방법 중, 확장 칼만 필터(EKF: Extended Kalman Filter)를 사용하는 방법이 가장 많이 연구되었으며 지금까지도 다수의 이동 로봇 응용에 이용되고 있다. 그러나 넓은 영역의 지도 작성을 할 때, 다수의 특징점 등록으로 인한 계산량이 증가하는 단점이 있다. EKF의 단점을 극복하는 알고리즘으로 Murphy는 Rao-Blackwellized particle filter (RBPF) [2]를 이용하는 방법을 제안하였으며, 다수의 연구자들에 의해 SLAM에 응용되었다. RBPF 기반의 SLAM은 사후 확률분포(posterior probability distribution)를 로봇 위치와 지도로 분해하여 두 사이의 상관관계를 없앴으로써 지도의 상태를 갱신하는 계산량을 줄일 수 있으며, 재샘플링(resampling) 과정을 통하여 잘못된 data association에 대응할 수 있는 장점이 있다[3,4]. 따라서 넓은 공간의 지도 작성이 요구될 때 RBPF 기반의 SLAM 기법을 이용하는 연구가 활발하다.

로봇의 이동 환경에 대한 정보는 주로 로봇에 부착된 거리 센서 (Laser, Sonar, IR 센서 등)를 이용하여 획득하는 방법과 비전 센서를 이용하여 영상 처리를 통하여 정보를 획득하는

방법이 있다. 최근에는 추출 가능한 정보량과 가격 측면에서 거리 센서보다 우월성을 가지는 비전 센서를 이용한 SLAM 기법들이 많이 연구되고 있다.

MonoSLAM 기법은 단일 카메라를 이용하여 좁은 공간에서의 카메라 궤적과 랜드마크 위치를 실시간으로 추정한다 [5]. 카메라는 이동 로봇에 고정되어 있지 않고 손으로 제어되며, 빠른 카메라 모션 후에도 카메라 위치를 지속적으로 예측하는 장점이 있다.

스테레오 비전과 SIFT 특징점을 이용한 SLAM기법 연구도 활발하게 진행되고 있다[6]. 두 장의 영상에서 각각 추출한 SIFT 특징점을 매칭시킴으로써 특징점들의 3D 좌표를 계산하고 지도에 등록시킨다. SIFT를 이용하면 다수의 특징점 추출이 가능하며 SLAM 성능 향상에 도움을 주는 장점이 있다.

CV-SLAM기법은 천장을 향하는 단일 카메라에서 얻은 영상을 이용하여 천장에 존재하는 특징점과 선분을 지도에 등록한다[7]. 카메라가 천장을 향해 있으므로 전방 카메라에서 얻은 영상의 경우보다 상대적으로 동적 물체에서 특징점이 추출되는 경우가 적어 정적인 환경에서의 특징점 추출을 가정할 수 있다.

본 논문에서는 CV-SLAM 기법과 같이 천장을 향하는 카메라를 이용하여 실내 복도 영역에서의 SLAM을 연구하였다. 좀 더 간결한 지도 표현을 위해 지도에 등록되는 랜드마크의 수를 줄임으로써 계산량을 적게 하였고, CV-SLAM과는 달리 천장과 벽면에서 추출되는 수평 선분만을 이용한 SLAM 기법을 제안하였다[8].

벽면에서 불안정적으로 추출(비연속적 추출)되는 수평 선분들을 제거하기 위하여 수평 선분 위의 점들이 가지는 에지 방향(edge orientation)의 분포를 이용한다. 2D 이미지에서 추출한 선분을 3D지도상에 등록하기 위해서는 수평 선분의 높이를 알아야 하며, 이를 위하여 선분 상의 코너를 추출 후 연속적인 추적을 통하여 높이를 예측한다. 현재 관측된 선분과 지도상에 등록된 선분의 매칭을 위하여 두 선분이 이루는 각,

* 책임저자(Corresponding Author)

논문접수: 2009. 10. 20., 수정: 2009. 11. 2., 채택확정: 2009. 11. 13.

안수용, 강정관, 이래경, 오세영: 포항공과대학교 전자전기공학과 (grasshop@postech.ac.kr/narool@postech.ac.kr/lacklee@postech.ac.kr/syoh@postech.ac.kr)

※ 본 연구는 (포항공과대학교 고품질로봇비전 연구센터를 통한)지식경제부/한국산업기술진흥원 융복합형로봇전문인력양성의 지원으로 수행되었음.

※ 상기 논문은 KACC2009 학술대회에서 초안이 발표되었습니다.

거리, 결합정도를 계산하여 data association을 수행한다. SLAM 수행 중, 천장의 동일한 수평 선분이 서로 다른 선분으로 등록되는 경우가 있는데, 등록된 선분들의 유사도를 주기적으로 계산하여 유사한 수평 선분들을 통합함으로써 계산량을 줄이고, 간결한 지도 표현을 추구한다.

본 논문은 다음과 같이 구성되어 있다. II 장은 시스템 구성 및 알고리즘 흐름도를 보여주며, III 장에서는 이미지에서의 수평 선분의 추출과 추적 방법을 다룬다. IV 장은 수평 선분의 상태를 초기화하는 방법에 대하여 다루며, V 장은 제안한 수평 선분 기반 RBPF-SLAM의 구조를 기술한다. VI 장에서는 수평 선분 최적화를 통한 간결한 지도 표현을 다루며, 마지막 VII 장은 실험 결과를 보여준다.

II. 시스템 구성

본 연구에서 사용된 이동 로봇 플랫폼은 Pioneer 3 DX (그림 1(a))이다. 지면으로부터 약 90cm 높이에 카메라를 장착하여 모든 실험을 진행하였다. 150도 이상의 화각 (Field of View: FOV)을 가지는 어안렌즈를 이용함으로써 낮은 천장 환경에서도 천장의 수평 선분이 관측 가능하도록 하였다(그림 1(b)). 그림 2는 전체 알고리즘 블록 다이어그램을 나타낸다.

III. 수평 선분의 추출 및 추적

1. 수평 선분의 추출

어안렌즈를 장착한 카메라로부터 얻은 영상은 왜곡이 심하기 때문에 이 영상에서 선분을 추출하는 것은 어렵다. 따라서, 먼저 카메라 내부 파라미터(intrinsic parameter)와 왜곡

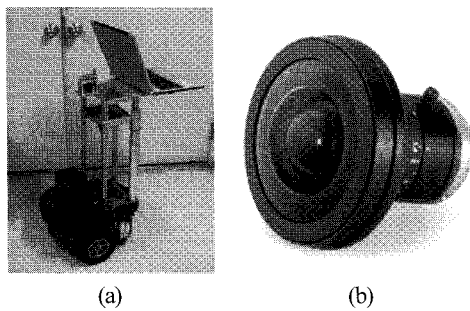


그림 1. (a) 로봇 플랫폼. (b) 어안렌즈.
Fig. 1. (a) Mobile robot platform. (b) Fish eye lens.

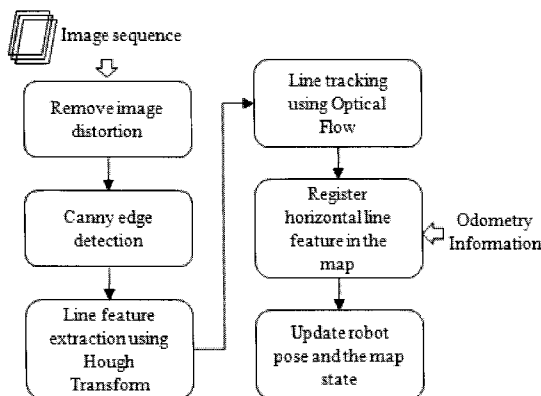


그림 2. 전체 알고리즘 블록 다이어그램.
Fig. 2. Block diagram for overall algorithm.

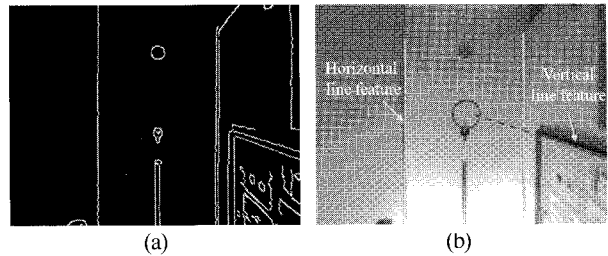


그림 3. (a) 에지 영상. (b) 추출된 수평 및 수직 선분.
Fig. 3. (a) Edge image. (b) Horizontal and vertical line features.

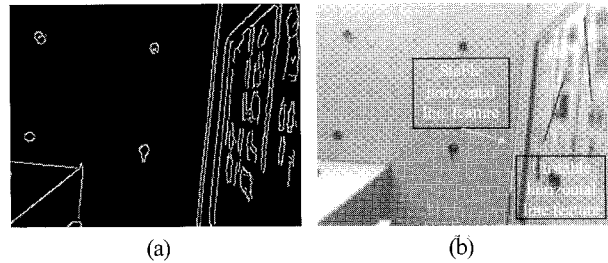


그림 4. (a) 에지 영상. (b) 불안정한 수평 선분의 추출.
Fig. 4. (a) Edge image. (b) Extraction of unstable horizontal line feature.

계수(distortion coefficient)를 구한 후 영상의 왜곡을 보정하는 과정이 필요하며, 본 논문에서는 [9]의 방법을 이용하여 왜곡을 보정하였다.

보정된 영상에서 Canny edge detector를 이용하여 에지 영상을 얻고, Hough transform을 적용하여 선분을 추출한다. 추출된 선분에는 그림 3과 같이 수평 선분 이외에 수직 선분(지면 혹은 천장과 수직)도 같이 추출되지만 수직 선분의 연장선은 영상의 중심을 지난다는 사실에 기반하여 간단히 제거 가능하다[7]. 본 논문에서는 천장 및 벽면에 존재하는, 연속적이면서도 안정적으로 추출되는 수평 선분을 이용한 SLAM 구현에 목적이 있으므로 그림 4와 같이 벽면에서 주로 발생하는 불안정한 수평 선분들을 제거할 필요가 있다(Hough transform을 이용한 에지 영상에서의 선분 추출은 흰색 픽셀의 Voting에 기반하기 때문에 예상하지 못한 곳에서 선분이 추출되는 경우가 있음).

불안정한 수평 선분들을 제거하기 위하여 본 논문에서는 선분 상의 점에서의 에지 방향(edge orientation)의 분포를 이용하였다. 추출된 수평 선분 위의 점을 일정 간격으로 샘플링 한 후, 그 점에서 아래와 같은 마스크를 적용한다.

$$P_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, P_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad (1)$$

k번째 점에서 마스크를 적용하여 얻은 값을 각각 p_x, p_y 라 하면 에지 방향은 다음과 같이 구할 수 있다.

$$\theta_k = \tan^{-1} \left(\frac{p_y}{p_x} \right) \quad (2)$$

수평 선분 상의 n개의 점에서 에지 방향을 구한 후, 그것의 표준 편차를 살펴보았을 때 미리 정의된 임계값 이하이면

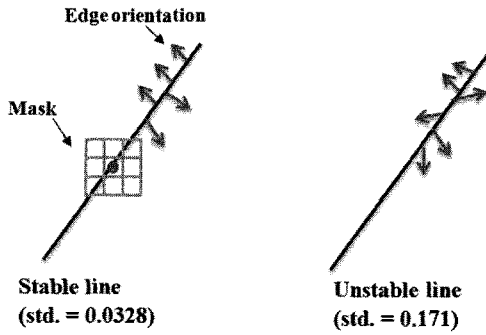


그림 5. 안정한 선분과 불안정한 선분의 에지 방향 분포.
 Fig. 5. Distribution of edge orientation for stable and unstable line features.

이 선분은 안정한 수평 선분이라고 판단할 수 있으며, 그렇지 않으면 불안정한 수평선분으로 간주하여 제거하게 된다(그림 5).

2. 수평선분의 추적

2D 영상에서 안정한 수평 선분을 추출 후, 3D지도 상에 등록시키기 위해서 연속적인 관측이 필요하다. 따라서 추출된 수평 선분을 추적하는 과정이 요구되며, 본 논문에서는 추출된 선분 상의 샘플링 된 점에 optical flow 기법과 Random sample consensus (RANSAC)를 적용하여 선분 추적을 구현하였다.

그림 6(a)에서 보듯이, t-1시점에서 추출된 선분에서 일정한 간격으로 점들을 샘플링한다. 이렇게 샘플링 된 점에 optical flow 기법을 적용하여 t 시점의 영상에서 어느 위치에 존재하게 될 것인지 예측한다. RANSAC을 이용하여 선분 파라미터를 구하고, 예측된 점들이 선분을 이루는지 조사한다(그림 6(b)). t 시점에서 계산된 선분과 각 점들 사이의 거리의 제곱의 합이 일정값 이상을 가지게 되면 선분의 추적이 실패한 것으로 간주한다. 이렇게 RANSAC으로 계산된 선분 파라미터는 t 시점의 영상에서 추출한 선분 파라미터와 정확히 일치하지 않으므로 추적 성공 여부에만 사용되며, t 시점의 영상에서 Hough transform을 이용하여 다시 정확한 수평 선분을 추출 후 추적할 선분 상의 점들을 재설정한다(그림 6(c)).

현재 추적되고 있는 수평 선분의 수가 3개보다 작으면 현재 영상의 전 영역에서 Hough transform을 이용하여 선분을 추출하고 이 중에서 추적되고 있지 않는 선분들을 새로운 추

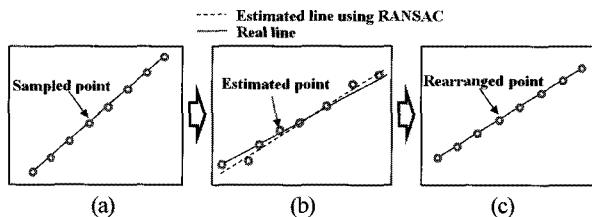


그림 6. (a) t-1 영상에서 추출된 수평 선분 위의 샘플링 된 점들. (b) RANSAC을 통한 선분 테스트. (c) t 영상에서 추적할 수평 선분 위의 점 재설정.

Fig. 6. (a) Sampled points from the line feature extracted at t-1 step. (b) Line test using RANSAC. (c) Rearrangement of sampled points for line tracking at t step.

적 대상으로 등록하게 된다. 추적되고 있는 선분의 수가 3개 이상이면 계산량을 줄이기 위하여 현재 추적 중인 선분들만 지속적으로 추적하게 된다.

IV. 수평 선분의 상태 초기화

수평 선분을 지도상에 등록시키기 위해서 수평 선분의 상태 $\mathbf{l}=(r, \varphi, z)$ 를 구해야 하며 앞서 말했듯이 이것은 관측된 수평 선분의 연속적인 추적을 통하여 구할 수 있다. 본 논문에서는 연속적으로 관측된 수평 선분의 파라미터를 이용한 초기 상태 계산 방법과 수직 선분과 수평 선분의 교점 추적을 통한 계산 방법을 구현한다.

1. 수평 선분의 관측 모델

수평 선분의 관측 모델은 다음과 같다.

$$h(\mathbf{x}_r(t), \mathbf{l}) = \begin{bmatrix} \rho \\ \theta \end{bmatrix} = \begin{bmatrix} (-r + x_r \cos \varphi + y_r \sin \varphi) \frac{f}{z} \\ \varphi - \theta_r - \frac{\pi}{2} \end{bmatrix} \quad (3)$$

$\mathbf{x}_r(t) = (x_r, y_r, \theta_r)$ 은 t시점에서의 로봇의 위치와 방향 상태를 나타내며, $\mathbf{l} = (r, \varphi, z)$ 는 수평 선분의 상태를 나타낸다. (ρ, θ) 는 영상에 투영된 수평 선분의 원점에서의 거리와 방향을 의미하며, f 는 어안렌즈의 초점거리이다(그림 7). 이 모델과 수평 선분의 관측 데이터를 이용하면 수평 선분의 상태를 간단히 구할 수 있다.

2. 관측 데이터를 이용한 수평 선분의 초기 상태 계산

연속적인 영상에서 수평 선분이 n번 추적 가능했다고 가정할 때, 수평 선분의 3D 상태 $\mathbf{l} = (r, \varphi, z)$ 는 다음과 같이 구할 수 있다.

1. φ 는 (3)식에 의해 아래와 같이 독립적으로 계산 가능하다.

$$\varphi_m = \theta_m + \theta_{r_m} + \frac{\pi}{2} \quad (4)$$

아래 첨자 m은 n번 관측 후의 평균값을 의미한다.

2. (r, z) 는 n번 관측 데이터 $(\rho_{i,m}, \theta_{i,m})$ 와 최소자승법(least square method)을 이용하여 다음과 같이 구할 수 있다.

$$\begin{bmatrix} f \\ \vdots \\ f \end{bmatrix} \begin{bmatrix} \rho_1 \\ \vdots \\ \rho_n \end{bmatrix} \begin{bmatrix} r \\ z \end{bmatrix} = \begin{bmatrix} (-x_{r_1} \cos \varphi_m - y_{r_1} \sin \varphi_m) f \\ \vdots \\ (-x_{r_n} \cos \varphi_m - y_{r_n} \sin \varphi_m) f \end{bmatrix} \quad (5)$$

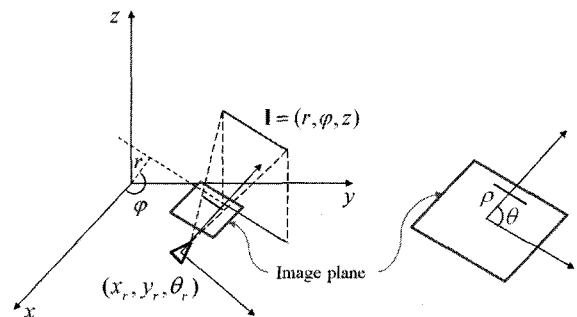


그림 7. 수평 선분의 관측 모델.

Fig. 7. Observation model for horizontal line feature.

위의 식은 $\mathbf{Ax} = \mathbf{b}$ 의 형태로 나타낼 수 있으며 따라서 유사역행렬 (Pseudo-inverse matrix)을 이용하면 $\mathbf{x} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}$ 의 해 (r, z) 를 구할 수 있다.

3. 수평 및 수직 선분의 교차점을 이용한 수평 선분의 초기 상태 계산

로봇이 관측된 수평 선분과 평행하게 움직이고 있다면 위의 방법으로 수평 선분의 초기 상태를 계산할 경우 부정확한 결과를 얻을 수 있다. 식 (5)에서 연속적으로 관측되는 수평 선분의 ρ 값이 동일하므로 행렬 \mathbf{A} 의 rank는 1이 된다. 이것은 수평 선분의 상태 계산에 도움이 되지 않는 여분의 정보를 뜻한다. 로봇이 복도 환경을 주행할 때 종종 이런 경우가 발생한다.

이 경우, 수평-수평 또는 수평-수직 선분의 교차점을 계산한 후 연속적으로 추적하여 교차점의 높이를 구한다. 교차점은 optical flow를 이용하여 추적하기 쉬우며 선분 위에 존재하기 때문에 그 높이를 선분의 높이로 간주할 수 있다(그림 8). 추적 가능한 교차점을 판단하기 위하여 교차점 주위의 픽셀 정보를 이용하여 아래의 행렬을 구성한다.

$$\mathbf{M} = \begin{bmatrix} \sum_w I_x^2 & \sum_w I_x I_y \\ \sum_w I_x I_y & \sum_w I_y^2 \end{bmatrix} \quad (6)$$

I_x, I_y 는 x, y축 방향의 픽셀값의 차이를 나타내며 W 는 교차점 주위의 3 x 3픽셀 영역을 의미한다. 위의 행렬 \mathbf{M} 의 고유값 (eigenvalue) λ_1, λ_2 가 아래의 조건을 만족하면 교차점은 추적 가능하다고 간주된다.

$$\min(\lambda_1, \lambda_2) > \lambda_0 \quad (7)$$

λ_0 는 미리 정의된 임계값이다. 연속적으로 추적된 교차점을 최소자승법을 이용하여 다음과 같이 높이를 계산할 수 있다.

$$\begin{bmatrix} f \sin \theta_{r1} & -f \cos \theta_{r1} & u_0 - u_1 \\ -f \cos \theta_{r1} & -f \sin \theta_{r1} & v_0 - v_1 \\ \vdots & \vdots & \vdots \\ f \sin \theta_{rm} & -f \cos \theta_{rm} & u_0 - u_n \\ -f \cos \theta_{rm} & -f \sin \theta_{rm} & v_0 - v_n \end{bmatrix} \begin{bmatrix} l_x \\ l_y \\ l_z \end{bmatrix} = \begin{bmatrix} x_{r1} f \sin \theta_{r1} - y_{r1} f \cos \theta_{r1} \\ -x_{r1} f \cos \theta_{r1} - y_{r1} f \sin \theta_{r1} \\ \vdots \\ x_{rm} f \sin \theta_{rm} - y_{rm} f \cos \theta_{rm} \\ -x_{rm} f \cos \theta_{rm} - y_{rm} f \sin \theta_{rm} \end{bmatrix} \quad (8)$$

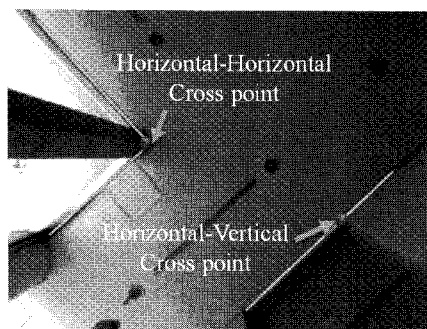


그림 8. 수평-수평 선분 및 수평-수직 선분이 만나서 이루는 교차점.

Fig. 8. Cross points by horizontal-horizontal and horizontal-vertical lines.

(u_0, v_0) 는 주점(principal point)이며, (u_n, v_n) 은 n번째 추적된 영상 내의 교차점의 좌표이다. 이전과 같이 유사역행렬을 이용하여 교차점의 3D 좌표 (l_x, l_y, l_z) 를 구할 수 있으며, l_z 가 바로 수평 선분의 높이가 된다.

V. RBPF 기반 SLAM

1. RBPF 기반 SLAM의 개요

SLAM을 위한 대부분의 필터링 알고리즘은 사후 확률 분포 $p(x_{1:t}, L | u_{1:t}, z_{1:t})$ 에 기반하고 있다. $x_{1:t}, u_{1:t}, z_{1:t}$ 는 1에서 t 시점까지의 로봇 상태, 컨트롤 입력, 관측값을 나타내며, L 은 지도상에 등록된 랜드마크를 의미한다. 위의 사후 확률 분포에 Rao-Blackwellization을 적용하면 아래 식과 같이 로봇 상태와 랜드마크를 서로 분리하여 필터링 가능하다.

$$p(x_{1:t}, L | u_{1:t}, z_{1:t}) = p(x_{1:t} | u_{1:t}, z_{1:t}) \prod_{i=1}^N p(l_i | x_{1:t}, u_{1:t}, z_{1:t}) \quad (9)$$

위의 분해로 인하여 총 1+N개의 필터가 사용되는데, 1개는 로봇 상태 예측을 위한 필터이며, N개는 랜드마크 상태 예측을 위한 필터이다. RBPF 기반 SLAM에서 로봇 상태 예측을 위하여 파티클 필터를 이용하며 랜드마크 상태 예측을 위하여 주로 확장형 칼만 필터를 이용한다.

2. 수평 선분 개수를 고려한 파티클 가중치 계산

모션 모델을 통하여 새로운 로봇 상태 샘플을 선택한 후, 파티클의 가중치를 다음과 같이 계산한다.

$$\omega_t^{(i)} = \frac{\text{target distribution}}{\text{proposal distribution}} = \eta p(z_t | x_t^{(i)}, u_{1:t}, z_{1:t}) \quad (10)$$

η 는 정규화 상수이다. 본 논문에서는 각 파티클이 가지는 수평 선분의 개수도 함께 고려하여 가중치를 계산하는 방법을 택한다. 실제 환경과 일치하지 않는 지도는 실제 지도보다 더 많은 수평 선분을 가지고 있으므로 선분의 개수가 적은 파티클에 더 큰 가중치를 부여하는 방식이다. 마지막으로, 파티클 가중치는 최대 및 최소 가중치 차이를 완화하기 위하여 g^* 함수를 이용하여 다시 계산된다.

$$\Delta^{(i)} = (\beta^{(i)} - \beta_{min})^2, \quad \tilde{\omega}_t^{(i)} = \omega_t^{(i)} \exp(-\Delta^{(i)} / \sigma_o^2),$$

$$g(x) = 1 / (1 + c_1 \exp(-c_2(x - 0.5))) + c_3, \quad (11)$$

$$\hat{\omega}_t^{(i)} = g(\tilde{\omega}_t^{(i)} / \tilde{\omega}_{max})$$

$\beta^{(i)}, \beta_{min}$ 은 각각 i번째 파티클이 가지는 수평 선분의 개수와 전체 파티클 중 수평 선분의 최소 개수이다. σ_o 는 가중치 감소의 정도를 결정하는 상수이며, $\tilde{\omega}_{max}$ 는 파티클 가중치 중 최대값을 의미한다. c_1, c_2, c_3 는 실험적으로 결정된 상수이다.

3. Time-stamp 기반의 재샘플링 방법

재샘플링은 파티클 분포를 원하는 분포로 유지시키기 위한 중요한 단계이다. 기본적인 파티클 필터 알고리즘은 매회 파티클의 가중치에 따라서 재샘플링을 수행하지만 잦은 재샘플링은 파티클의 다양성을 줄일 수 있으므로 재샘플링 시기를 결정하는 전략이 중요하다.

본 논문에서는 재샘플링 시기 선택에 있어서 두 가지 방법을 이용하는데 그것은 주기적인 재샘플링과 time-stamp기반의 재샘플링(time-stamp based resampling)이다(그림 9). 주기적

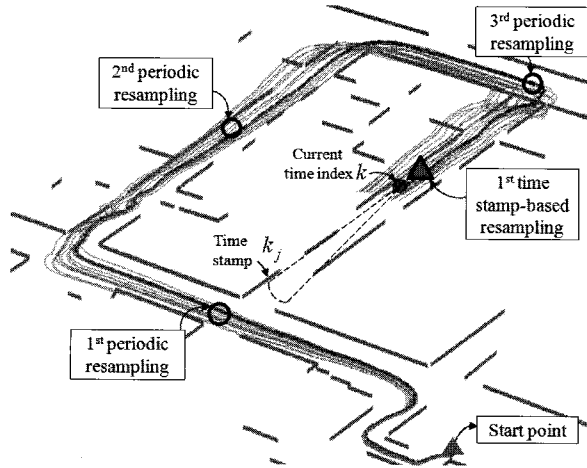


그림 9. 주기적인 재샘플링과 Time-stamp 기반 재샘플링.
Fig. 9. Periodic and time-stamp based resampling.

인 재샘플링(periodic resampling)은 자주는 아니지만 지정된 시간이 경과된 후 주기적으로 일어나며, 이것은 파티클의 발산을 막는 역할을 한다. 따라서 넓은 영역 주행 후의 loop closing를 위한 수동적인 재샘플링 방법이라 볼 수 있다. Time-stamp기반의 재샘플링은 loop closing이 일어난 후 수행된다는 점에서 [10]의 방법과 유사하다. 로봇이 새로운 수평 선분 l_j 을 지도에 등록시킬 때마다 time stamp k_j 가 l_j 에 부여되며, 현재 관측된 수평 선분이 지도에 등록된 l_j 와 매칭될 경우 현재 time stamp k 와 k_j 의 차를 계산한다. 둘의 차이가 미리 정의된 k_{th} 보다 크면 loop closing으로 간주하여 재샘플링을 수행하며, 그렇지 않으면 수평 선분의 재관측(re-observation)으로 판단한다. 이 방법은 특별한 loop closing 탐지 방법 없이 loop closing과 재관측을 간단히 구분하며, 또한 재샘플링의 지연 정도를 조절한다.

VI. 수평 선분의 최적화 방법

로봇은 미지의 환경을 주행하면서 추출한 수평 선분을 지도에 등록하게 된다. 하지만 로봇의 위치오차로 인하여 동일한 수평 선분이 지도에 등록될 때는 다른 선분으로 등록될 가능성이 있다(그림 10). 따라서 서로 다르게 등록된 2개의 선분을 하나로 병합하는 과정이 필요하다. 이 장에서는 두 수평 선분의 매칭정도를 판단하는 방법과 covariance intersection을 이용한 수평 선분의 상태 병합에 대해 다룬다.

1. Data Association

3D지도상의 두 선분이 2D영상좌표계로 투영되어 매칭정

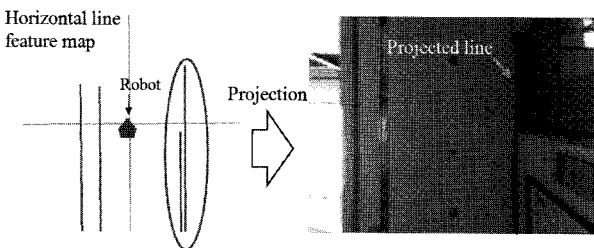


그림 10. 지도에 등록된 수평 성분의 2D 영상 투영.
Fig. 10. Projection of registered line feature on 2D image plane.

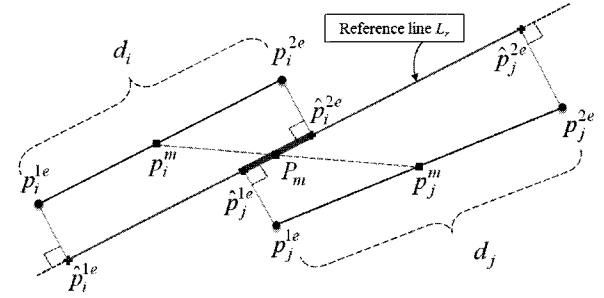


그림 11. 두 선분의 겹침 정도와 거리 계산.
Fig. 11. Overlapping factor and distance between two line features.

도가 계산된다. 2D영상으로 투영된 선분은 $l_i = \{\rho_i, \alpha_i, p_i^{1e}, p_i^{2e}, p_i^m, d_i\}$ 의 파라미터를 가진다. ρ_i, α_i 는 영상중심에서 선분까지의 거리와 선분의 각도이며, $p_i^{1e}, p_i^{2e}, p_i^m$ 는 선분의 양 끝점의 좌표와 그 중심좌표이다. d_i 는 선분의 길이를 나타낸다. 먼저, 두 선분의 겹침정도 Q 를 계산하기 위해 기준선(reference line) L_r 의 파라미터를 아래와 같이 설정한다.

$$\alpha_g = \frac{d_i \alpha_i + d_j \alpha_j}{d_i + d_j}, \quad p_m = \frac{d_i p_i^m + d_j p_j^m}{d_i + d_j} \quad (12)$$

두 선분의 양 끝점들을 위에서 설정한 기준선에 투영하여 얻은 점들을 각각 $\hat{p}_i^{1e}, \hat{p}_i^{2e}, \hat{p}_j^{1e}, \hat{p}_j^{2e}$ 이라 하면, 다음과 같이 두 선분의 겹침 정도를 계산할 수 있다(그림 11).

$$Q = \frac{ED(\hat{p}_i^{1e}, \hat{p}_i^{2e}) + ED(\hat{p}_j^{1e}, \hat{p}_j^{2e})}{d_r} \quad (13)$$

d_r 은 기준선의 길이이며, ED 는 Euclidean 거리를 계산하는 함수이다. 위와 같이 Q 를 정의했을 때 (0, 2]사이의 값을 가지며, 1을 기준으로 두 선분의 겹침 정도를 판단한다. Q 는 두 선분이 떨어져있는 거리에 대해서는 정보를 주지 못하므로 다음과 같이 두 선분 사이의 거리를 구한다.

$$D_{ij}^p = \sum_{k=1}^2 \frac{ED(p_i^{ke}, \hat{p}_i^{ke}) + ED(p_j^{ke}, \hat{p}_j^{ke})}{2} \quad (14)$$

위의 두 조건과 선분의 각도 유사도를 이용하여 두 선분의 매칭을 결정한다.

$$|\alpha_i - \alpha_j| < \alpha_{th}, \quad Q < Q_{th}, \quad D_{ij}^p < D_{th} \quad (15)$$

$\alpha_{th}, Q_{th}, D_{th}$ 는 미리 정의된 임계치이다.

2. 공분산 교차(covariance Intersection) 방법을 이용한 수평 선분 최적화

두 선분이 매칭되었다면 두 선분의 상태 병합을 통하여 하나의 선분으로 나타낸다. 서로 다른 두 상태를 병합하기 위하여 본 논문에서는 공분산 교차방법을 이용한다[11]. 두 선분의 상태와 공분산을 각각 $(\mu_i, \Sigma_i), (\mu_j, \Sigma_j)$ 라고 할 때 병합된 상태와 공분산은 다음과 같이 계산할 수 있다.

$$\hat{\Sigma}^{-1} = \omega \Sigma_i^{-1} + (1 - \omega) \Sigma_j^{-1} \quad (16)$$

$$\hat{\mu} = \hat{\Sigma}(\omega \Sigma_i^{-1} \mu_i + (1 - \omega) \Sigma_j^{-1} \mu_j)$$

ω 는 0과 1사이의 값을 가지는 파라미터인데, 보통 공분산의 trace가 최소가 될 때의 값으로 설정한다. 그러나 지도에 등록된 수평 선분들은 상호 연관되어 있지 않으므로 ω 를 0.5로 두는데, 이것은 상호 연관되어있지 않은 두 상태의 최적의 병합이며 칼만 필터 업데이트와 일치한다.

VII. 실험 결과

실험은 부천 테크노파크 1층 전시관과 POSTECH LG 연구동 3층 복도 환경에서 이루어졌다.

1. 실험 환경

- 테크노파크 1층 전시관: 44m x 28m의 공간으로 천장이 특정한 격자 무늬로 되어있어 다수의 선분 특징점의 추출이 가능하다. 카메라와 천장 사이의 거리는 2.0m이며 복도영역에서 출발하여 홀 영역을 거쳐서 다시 원위치 주위로 돌아오는 경로를 취한다. 수동 조작을 통하여 평균 0.2m/s의 속도로 총 125m를 주행하였으며, 주행 중 얻은 이미지의 개수는 2,251개이다. 그림 12(a)는 로봇이 주행한 환경의 2D지도를 대략적으로 보여준다.

- POSTECH LG연구동 3층 복도: 54m x 36m의 공간으로 복도의 폭은 2m, 천장은 지면으로부터 2.6 m 떨어져있다 (천장과 카메라 사이는 1.7m 떨어져있음). 로봇은 수동 조작을 통하여 0.4~0.5m/s 의 속도로 총 280 m를 주행하였으며, 주행 중 얻은 이미지의 개수는 3,840개이다. 그림 12(b)는 로봇이 주행한 환경의 2D지도를 보여준다.

2. 실험 결과

- 테크노파크 1층 전시관 실험 결과: 그림 13은 본 논문에서 제안된 방법을 이용하여 지도 작성을 수행하여 2D 평면에 나타낸 결과이다. 연두색 실선은 odometer로 측정한 로봇의 경로인데 페루프를 이루지 않고 발산하는 것을 볼 수 있다. 빨간색 점선은 제안된 알고리즘을 이용하여 로봇의 경로를 예측한 것이며, 출발지점과 도착지점의 위치가 일치하는 것을 확인할 수 있다. 이것은 예측된 로봇의 경로와 실제 경로가 일치함을 의미한다. 파란색 가는 실선은 지도에 등록된 수평 선분을 나타낸다.

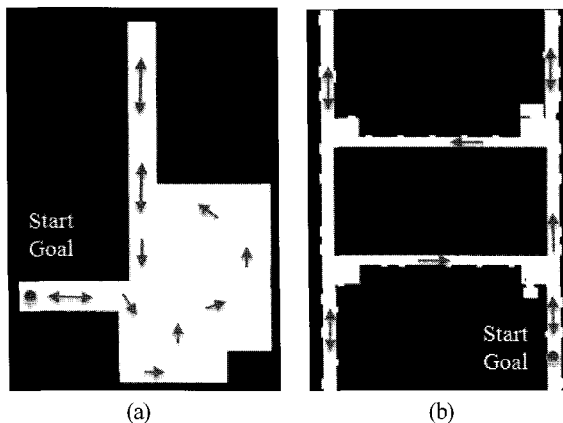


그림 12. (a) 부천 테크노파크 1층 전시관 지도. (b) POSTECH LG 연구동 3층 복도 지도.

Fig. 12 (a) Simplified 2D map of Bucheon.Technopark. (b) Simplified 2D map of 3rd floor of POSTECH LG research center.

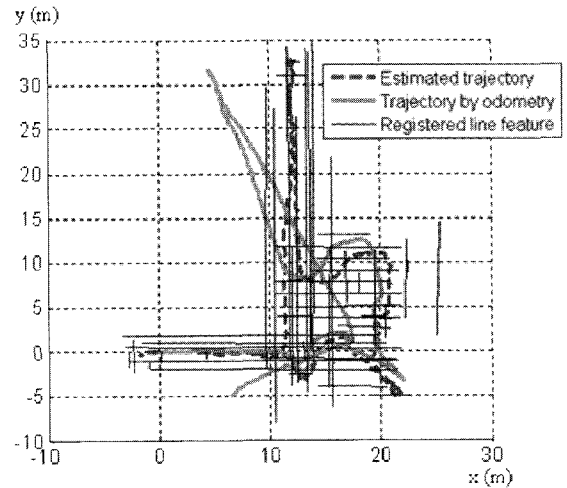


그림 13. 테크노파크 실험 결과 - 예측된 로봇 경로와 등록된 수평 선분의 2D 표시.

Fig. 13. Experimental result at technopark - Estimated robot trajectory and 2D representation of registered horizontal line features.

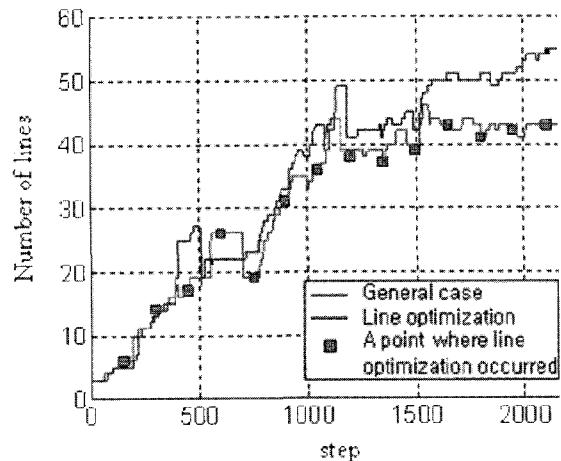


그림 14. 선분 최적화로 인한 등록된 선분 개수의 변화.

Fig. 14. Number of registered line features by line optimization.

실험에 사용된 총 파티클의 수는 60개로서, 이것은 실시간 위치 인식 및 지도 작성에 큰 부담을 주지 않는다. 수평 선분 최적화 과정을 통하여 지도에 등록된 총 선분 개수는 43개이다(그림 14). 최적화 과정을 거치지 않으면 수평 선분이 중복으로 등록되는 경우가

발생하는데 이 실험의 경우 총 56개의 선분이 등록이 되었으며 최적화 과정을 거친 경우보다 13개의 선분이 더 많았다. 따라서 등록된 선분의 최적화 과정을 거치게 되면 계산량 및 데이터 저장 공간을 줄일 수 있으며 좀 더 간결한 지도의 표현이 가능하다. 최종적인 위치 오차 (출발점과 도착점의 위치 차이)는 평균 0.5m 이내로 측정되었다.

- POSTECH LG연구동 실험 결과: 그림 15는 복도 환경에서 지도 작성을 한 결과이다. 첫번째 실험과 마찬가지로 예측된 경로는 페루프를 이루고 있으며, 로봇 경로 양쪽으로 수평 선분이 등록되어 있다. 등록된 선분은 대부분이 천장에 존재하는 선분이며, 벽면에서 추출되는 선분 (문, 캐비닛 등으로

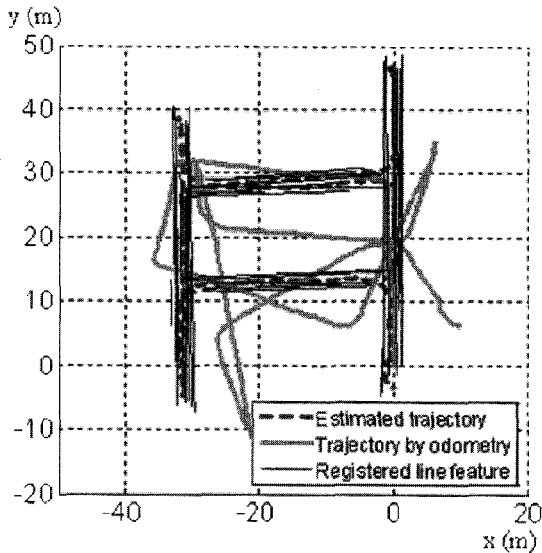


그림 15. POSTECH 실험 결과 - 예측된 로봇 경로와 등록된 수평 선분의 2D 표시.

Fig. 15. Experimental result at POSTECH - Estimated robot trajectory and 2D representation of registered horizontal line features.

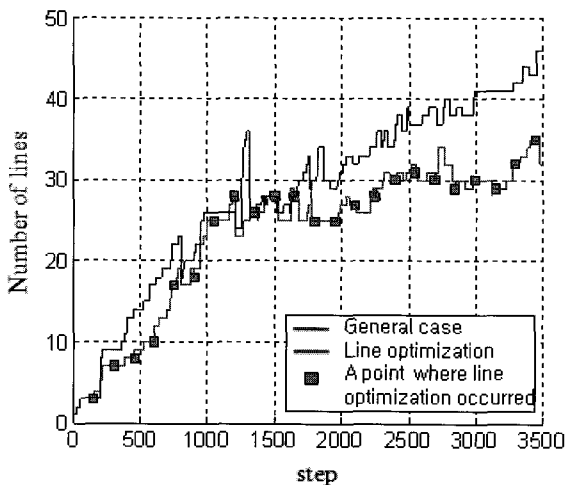


그림 16. 선분 최적화로 인한 등록된 선분 개수의 변화.

Fig. 16. Number of registered line features by line optimization.

인한 선분) 역시 등록되었다. 사용된 파티클의 수는 총 60개이며 등록된 선분의 개수는 32개이다. 그림 16은 등록된 수평 선분의 최적화로 인한 효과를 보여준다.

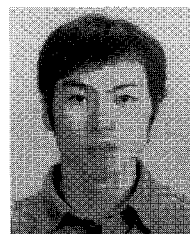
VIII. 결론

본 논문에서는, 천장을 향하는 카메라에서 얻은 영상에서 수평 선분을 추출하고 이 선분을 이용하여 지도를 작성하는 알고리즘이 제안되었다. 안정적인 선분만을 지도에 등록하기 위해 에지 방향을 이용하여 불안정한 선분을 제거하였고, optical flow를 이용하여 추출된 선분을 추적하였다. 최소자승법을 이용한 2가지 방법으로 선분의 초기 상태를 예측하고, 상태가 유사한 두 선분은 공분산 교차 방법으로 병합하여 좀더 간결한 지도 작성을 구현하였다. 부천 테크노파크와 POSTECH LG연구동에서 실제 실험을 수행하였고 Pentium

1.8GHz 환경에서 실시간 위치 인식 및 지도 작성이 가능함을 보였다.

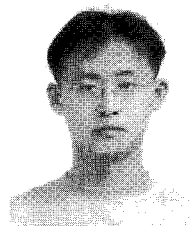
참고문헌

- [1] S. Thrun, W. Burgard, and D. Fox, "Probabilistic robotics," MIT Press, Cambridge, 2005.
- [2] A. Doucet, N. Freitas, K. Murphy, and S. Russell, "Rao-blackwellized particle filtering for dynamic bayesian network," *Proceedings of the 16th conference on Uncertainty in Artificial Intelligence*, pp. 176-183, 2000.
- [3] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "Fast-SLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges," in *Proc. Int. Joint Conf. Artif. Intell.*, vol. 18, pp. 1151-1156, 2003.
- [4] G. Grisetti, C. Stachniss, and W. Burgard, "Improving grid-based slam with Rao-Blackwellized particle filters by adaptive proposals and selective resampling," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pp. 2432-2437, April, 2005.
- [5] A. Davison et al, "MonoSLAM: Real-time single camera SLAM," *IEEE Trans. PAMI*, vol. 29, no. 6, pp. 1052-1067, June 2007.
- [6] R. Sim, P. Elinas, M. Griffin, and J. Little, "Vision-based SLAM using the Rao-Blackwellised particle filter," *IJCAI Workshop reasoning with Uncertainty in Robotics*, 2005.
- [7] W. Jeong and K. Lee, "Visual SLAM with line and corner features," *IROS*, pp. 2570-2575, 2006.
- [8] 안수용, 강정관, 이레경, 오세영, "넓은 실내 공간에서 수평 선분을 이용한 비전기반의 지도작성 및 위치인식," 한국자동제어학술회의 (KACC 2009), pp. 263-268, 2009.
- [9] Z. Zhengyou, "Flexible camera calibration by viewing a plane from unknown orientations," *ICCV*, pp. 666-673, 1999.
- [10] C. Kim and W. Chung, "Delayed resampling in a Rao-Blackwellized particle filtering SLAM for consistent loop closure," *IROS*, pp. 2084-2090, 2008.
- [11] S. Julier and J. Uhlmann, "Using covariance intersection for SLAM," *Robotics and Autonomous system*, vol. 55, no. 1, pp. 3-20, Jan. 2007.



안수용

2006년 포항공과대학교 전자전기공학과 졸업. 2006년~현재 포항공과대학교 대학원 전자전기공학과 통합과정 재학 중. 관심분야는 이동로봇, SLAM.



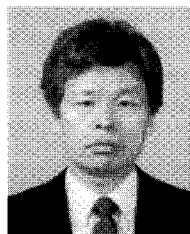
강정관

2002년 경북대학교 전자공학과 졸업. 2004년 포항공과대학교 대학원 전자공학과 졸업(공학석사). 2004년~현재 포항공과대학교 전자전기공학과 박사과정 재학 중. 관심분야는 신경망, 이동로봇, 지능자동차.



이 래 경

1999년 부산대학교 전기공학과 졸업.
2001년 부산대학교 대학원 전기공학과
졸업(공학석사). 2009년~현재 포항공과
대학교 전자전기공학과 박사과정 재학
중. 관심분야는 이동로봇, HRI, 지능제어.



오 세 영

1974년 서울대학교 전자공학과(공학사).
1978년 Case Western Reserve Univ. 전자
공학과(공학석사). 1981년 Case Western
Reserve Univ. 전자공학과(공학박사).
1981년~1984년 Univ. of Illinois 조교수.
1985년~1988년 Univ. of Florida 조교수.
1988년~현재 포항공과대학교 전자전기공학과 정교수. 관심분
야는 신경망, 진화연산, 이동로봇, 지능자동차, 지능시스템.