

이동정보를 배제한 위치추정 알고리즘

SIFT-Like Pose Tracking with LIDAR using Zero Odometry

김 지 수, 곽 노 준*
(Jee-Soo Kim¹ and Nojun Kwak^{1,*})

¹Department of Transdisciplinary Studies, Seoul National University

Abstract: Navigating an unknown environment is a challenging task for a robot, especially when a large number of obstacles exist and the odometry lacks reliability. Pose tracking allows the robot to determine its location relative to its previous location. The ICP (iterative closest point) has been a powerful method for matching two point clouds and determining the transformation matrix between the maps. However, in a situation where odometry is not available and the robot moves far from its original location, the ICP fails to calculate the exact displacement. In this paper, we suggest a method that is able to match two different point clouds taken a long distance apart. Without using any odometry information, it only exploits the features of corner points containing information on the surroundings. The algorithm is fast enough to run in real time.

Keywords: pose tracking, localization, odometry, uncertain robot pose

I. 서론

2011년 후쿠시마에 들이닥친 해일로 인해 원전이 붕괴되었고, 이로 인해 반경 20Km이내의 토지와 인근 해안이 방사능으로 크게 오염되는 사고가 발생하였다. 전문가들은 사고가 발생한 시점에 누군가 적절한 대처를 해주었다더라면 피해는 크게 경감되었을 것이라 판단했지만, 인간의 목숨을 담보로 해야 하기 때문에 특정 사람에게 강요할 수 있는 문제가 아니다. 이 역할을 로봇에게 시킨다면 윤리적 문제는 해결이 되므로, 기술적 문제를 해결해보고자 진행되었던 대회가 미국의 DARPA (Defence Advanced Research Projects Agency) Robotics Challenge이다. 통신이 원활하지 않은 상황에서 로봇이 걸어서 벨브를 잠그거나, 드릴로 구멍을 뚫는 것과 같은 과제를 수행하는 대회이다. 이를 위해서는 다른 기술도 중요하지만, 특히 로봇이 자신의 위치를 인식할 수 있는 SLAM (Simultaneous Localization and Mapping) 기술이 중요하다[1]. SLAM에서 거리정보로 주어진 두 점-구름(point cloud)의 변환행렬(transformation matrix)을 구하는 것은 매우 중요한 과정이고 이를 위치추정(pose tracking)이라 부른다. 이 과정에서 오차가 적어야 지도를 만드는 과정에서의 오차 역시 줄일 수 있다. 가장 흔히 사용하는 위치추정 알고리즘으로는 ICP (Iterative Closest Point)가 있다[11]. ICP는 점과 점간의 관계를 연산하는 방식(point to point)이기 때문에 초기정보가 주어지지 않으면 국소최저치에 수렴한다는 문제점이 있다. 따라서 일반적으로 위치추정은 추측항법(dead reckoning)에 크게 의존한다. 추측항법으로 가장

많이 사용하는 주행거리측정방법(odometry)은 미끄러짐이나 내부센서의 오차와 같은 요인으로 인해 오차가 계속해서 누적되는 어려움이 있다[2]. 이를 해결하기 위해 칼만 필터를 이용하여 내부센서에서 발생하는 오차를 외부센서(초음파센서, LIDAR 등)로 보정하는 노력이 많이 이루어졌다[3,4]. 이러한 노력의 일환으로 정확도를 높이기 위해 거리정보를 영상과 접목하여 사용하기도 한다[5,12]. 하지만 뚜렷한 비교대상(landmark)을 찾지 못하거나, 비교대상이 시점에 따라 가려지면 이 역시 사용이 다소 어려울 수 있다는 문제가 있다. 적당한 특징요소를 생성하여 위치추정을 시도한 방법들도 많이 제시되었다. 거리정보에서 얻은 점선 성분들을 이용하여 방향(orientation)에 강건하도록 만든 방법이 있으나 이 역시 추측항법에 크게 의존한다[6]. 또한 직선성분들을 허프 영역(Hough domain)으로 변환한 후에 스펙트럼을 만들어 이를 이용하여 회전과 변위를 찾는 방법도 제시되었다. 하지만 이 방법은 직선이 크게 우세한 상황을 가정해야 하며, 따라서 방해요소가 크면 잘 동작하지 않는 문제를 안고 있다[7].

2015년 DARPA robotic challenge는 재난로봇들이 주어진 임무를 수행하며 경합하는 대회였다. 대부분의 로봇들이 이 족보행을 하였으며, 통신이 극도로 불안정하여 짧은 시간 내에 로봇의 위치를 알아낼 방법이 필요하였다. 또한 이족보행 휴머노이드는 걸어가면서 생기는 불규칙적인 움직임 때문에 제어량을 정량화하기가 힘들어서 추측항법을 사용하면 오차가 크게 누적된다. 건물 전체와 같이 스케일이 큰 환경에서는 particle filter를 이용한 위치추정이 많이 사용되며, 성능도 우수하다[8]. 그러나 휴머노이드와 같이 이동하는데 많은 시간이 걸리며 실제 이동량이 적고, 제어량과 이동량에 오차가 크게 발생하는 경우에 particle filter의 성능은 보장받을 수 없다. 또한 지속적으로 다음위치를 제안하고 검증하는 과정을 반복하는 방식이므로 즉각적이지 못하다.

* Corresponding Author

Manuscript received July 28, 2016 / revised October 1, 2016 / accepted October 14, 2016

김지수, 곽노준: 서울대학교 융합과학기술부

(kimjiss0305@snu.ac.kr/nojunk@snu.ac.kr)

※ 본 연구는 문화체육관광부 및 한국콘텐츠진흥원의 2016년도 문화기술연구개발지원사업의 연구결과로 수행되었음.

그래서 이 논문에서는 주행정보가 없이 즉각적으로 위치 추정을 하는 방법인 SPT (SIFT-like Pose Tracking without odometry)를 제시한다. 영상처리에서 많이 사용하는 방식인 SIFT (Scale Invariant Feature Transform)는 이미지상에서 식별이 용이한 화소를 중심으로 화소들의 변화율(gradient)을 히스토그램으로 변환하여 식별자(descriptor)로 만든 방법이다[10]. 본 논문에서는 이 방법에 착안하여 실내 환경에서 하나의 LIDAR 센서로 찾아낸 모서리에 주변의 환경을 반영한 특징벡터(feature vector)를 부여하여 주행정보 없이도 특징기반 위치추정이 가능한 알고리즘을 고안했다. 제안하는 방법을 이용할 경우 사람이 지나가면서 기존 지형을 가리더라도 이를 무시하고 위치추정이 가능하다. 또한, 사무실이나 불규칙한 표면을 가진 물체들이 있는 실내 환경에서도 높은 정확도로 위치추정이 가능하다.

II. 위치추정 알고리즘

1. 모서리 탐색

로봇이 이동을 하며 LIDAR로 두 개의 점-구름을 만들었다고 가정하도록 하자. 이전 상태에서 얻은 점-구름에서 센서의 위치를 P_{ref} , 현재 상태에서 얻은 점-구름에서 센서의 위치를 P_{curr} 라 할 때, 위치추정 문제에서는 식 (1)과 같이 두 점-구름을 정확하게 일치시키는 회전과 변위를 구하고자 한다.

$$P_{curr} = P_{ref}R + T \quad (1)$$

이렇게 찾아진 R과 T는 로봇의 상대적 위치를 규명하는데 사용된다.

2. 모서리 탐색

본 논문에서 제시하는 SPT방법의 전체적인 흐름은, 모서리를 찾은 후에 그 모서리 주위의 환경을 descriptor정보로 부여하여 다른 점-구름 지도에서 같은 descriptor를 갖는 모서리를 찾아 대응시키는 것이다. 실내 환경에서 각진 모서리는 아주 좋은 특징점이기 때문에 우선적으로 각진 모서리를 찾는다. 거리센서(Hokuyo UTM-30LX)에서 거리정보가 담긴 data를 받아오고, 모든 점에서 RANSAC [8]을 이용하여 양방향으로 두 직선을 (fitting)찾는다. 이때 그 두 직선은 각을 이루게 되는데, 잡음을 줄이기 위해 먼저 비최소치 억제(Non-minimum suppression)를 수행하고 그 각도가 일정 값(th_{edge}) 이상인 지점들을 모서리로 정의한다. SPT 방법에서는 이렇게 찾아진 모서리들에게만 descriptor를 부여하므로 계산량을 대폭 줄일 수 있다.

3. Descriptor 생성

다음 단계는 찾아낸 모서리에 그 모서리만이 갖는 특수한 정보를 부여하는 것이다. 이를 위해 찾아낸 모서리마다 16개의 feature로 구성된 descriptor를 만든다. 영상처리분야에서 흔히 사용되는 SIFT에서는 이미지가 회전되었을 때를 대비하여 가장 큰 놈(norm)을 갖는 변화율을 기준 방향으로 잡고 이 방위를 기준으로 feature vector를 생성하는 방법을 사용한다. 같은 맥락으로 SPT에서도 기준이 될 방향을 먼저 정한 후에 feature vector를 생성한다. 실내 환경의 특성상 모서리는 면과 면이 교차하는 점으로 가정하고, 앞

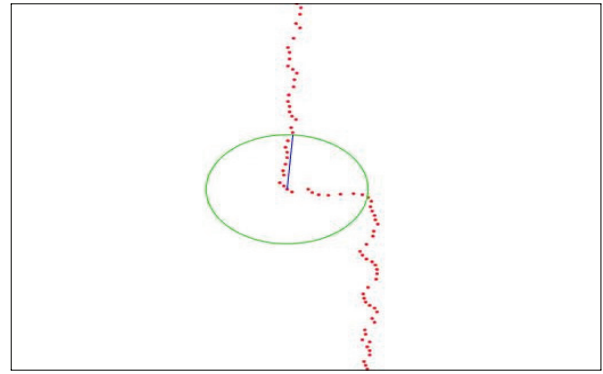


그림 1. RANSAC을 이용한 좌측 벽 탐색.

Fig. 1. Searching the left wall using RANSAC.

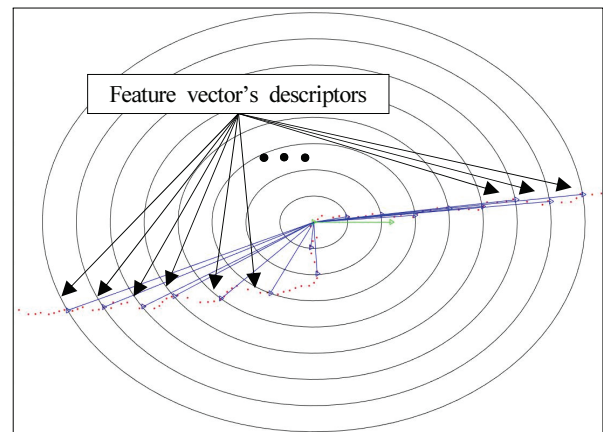


그림 2. Feature vector 생성과정.

Fig. 2. Process of generating a feature vector.

서 탐색된 모서리에서 그림 1에서와 같이 RANSAC을 이용하여 왼쪽 벽을 직선으로 근사한다. 그림 1에서 모서리로 판정된 점이 원의 중심이 되며, 이 점을 시작으로 한쪽방향에 있는 점들을 RANSAC에 사용한다. LIDAR에서 들어온 data는 센서가 위치한 지점을 중심으로 반시계방향으로 차례대로 수집되므로, 센서의 위치에서 바라봤을 때 오른쪽 벽에서 왼쪽 벽으로 값이 들어온다. 따라서 왼쪽 벽에 있는 점들은 모서리점의 data 이후에 들어온 data들에 해당한다. 이렇게 탐색된 직선방향이 (1,0) 방향을 향하도록 해당 지도의 점들을 전부 회전시킨 후에, 모든 점들까지의 거리를 계산하여, 좌우로 선을 따라가면서 0.35m를 8등분한 위치에서의 점들의 좌표 16개를 descriptor로 만든 것이 가장 좋은 결과를 내었다. 이렇게 하면 그 모서리의 반경 35cm 안에 있는 지형지물의 생김새에 대한 정보를 담게 된다. 더 많은 개수의 좌표를 사용하여 descriptor를 만들 수 있으나 16보다 더 많은 숫자의 feature vector는 성능을 더 이상 향상시키지 못했다. 또한 반경을 더 크게 잡으면 더 넓은 영역을 담아낼 수 있으나 실험결과 35cm가 다른 길이에 비해 최종 pose tracking결과의 변위오차가 가장 적게 나왔다. 이렇게 하여 $F_{(i)} = [P_1^{(i)}, P_2^{(i)}, \dots, P_{16}^{(i)}]$ ($P_k^{(i)} = (x_k^{(i)}, y_k^{(i)})$) 형태의 i번째 모서리에서의 32차원의 feature vector를 얻을 수 있다. 불규칙한 표면위에서 모서리라고 판단된 점들은 이

과정에서 불규칙한 방향을 대표벡터로 갖는다. 따라서 후에 불규칙한 descriptor를 만들게 되며, 강건한 속성을 잃게 된다. 그림 2는 이러한 과정을 통해 feature vector를 생성한 예이다.

4. Descriptor 비교

Descriptor를 만들었다면 이를 이용해 서로 다른 두 개의 점-구름 지도에서 이들을 비교하여 유사한 속성을 갖는 모서리들끼리 엮을 수 있다. 로봇이 움직이기 전에 LIDAR센서를 이용하여 찍은 점-구름을 기준 지도(reference map), 로봇이 이동을 하고 난 후에 찍은 점-구름을 지역 지도(local map)로 가정한다. Local map에 있는 한 모서리와 reference map에 있는 한 모서리의 각 feature vector의 오차를 구한다. 오차가 작다면 주변지리가 유사하다는 의미이므로, 같은 모서리라고 판단할 가능성이 커진다. 각 모서리에서의 feature vector를 Local map에서 $F_{(l)}$, reference map에서 $F_{(r)}$ 로 표현한다면 오차는 $(F_{(l)}-F_{(r)})$ 의 L2-norm으로 구한다. 이 오차값은 물리적으로는 해당 모서리들이 가지고 있는 지형들끼리 서로 얼마나 어긋나 있는지를 정량화한 값이라고 생각할 수 있다. 이렇게 비교하여 얻은 오차값이 임의로 정한 역치값 T_s 보다 작으면 강 유사 점대응(strong similarity pair)으로 정하고, 그 외에 쌍들 중 오차가 T_w 보다 작은 쌍들은 약 유사 점대응(weak similarity pair)으로 정한다. 그 이상의 오차를 갖는 쌍들은 전부 버린다. 이렇게 일치한다고 생각되어지는 모서리들끼리 쌍으로 짝을 지어줄 수 있고, 이들의 집합을 set S 라 한다. 이 때, weak similarity pair보다 strong similarity pair에 더 높은 가중치를 주고, 뒤에서 설명하는 변환행렬 탐색과정에서 이 가중치를 이용한다.

5. 변환행렬 탐색

유사한 descriptor를 갖는 모서리들로 짝을 지은 pair들의 집합 S 안에는 정확히 일치하는 점을 짝 지은 쌍도 있을 것이고, 잘못된 점을 짝 지은 쌍도 있을 것이다. 하지만 어느 것이 제대로 짝지어진 쌍인지 알 수 없다. 따라서 집합 S 에서 두 개의 쌍을 골라 이 두 쌍을 일치시키는 변환행렬을 구한 후에 다른 나머지 쌍들에서 이 변환행렬이 얼마나 정확한지를 점수로 매긴다. 설명을 위해 S 에서 임의로 고른 두 개의 pair가 각각 다음 식 (2)와 같이 P_1, P_2 라 한다.

$$\begin{aligned} P_1 &= [(x_p, y_p), (x'_q, y'_q)] \\ P_2 &= [(x_r, y_r), (x'_s, y'_s)] \end{aligned} \quad (2)$$

이들을 식 (1)에 넣으면 다음과 같은 식을 얻을 수 있다.

$$\begin{aligned} R \begin{bmatrix} x_p \\ y_p \end{bmatrix} + T &= \begin{bmatrix} x'_q \\ y'_q \end{bmatrix} \\ R \begin{bmatrix} x_r \\ y_r \end{bmatrix} + T &= \begin{bmatrix} x'_s \\ y'_s \end{bmatrix} \end{aligned} \quad (3)$$

여기에서 R 은 다음 식과 같이 회전행렬(rotation matrix)를 의미하고, T 는 변위행렬(translation matrix)를 의미한다.

$$\begin{aligned} R &= \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \\ T &= \begin{bmatrix} t_x \\ t_y \end{bmatrix} \end{aligned} \quad (4)$$

두 개의 쌍이 주어지면 θ 와 t_x, t_y 는 다음과 같이 쉽게 구할 수 있다.

$$\theta = \cos^{-1} \left(\frac{(x_1 - x_2)(x'_1 - x'_2) + (y_1 - y_2)(y'_1 - y'_2)}{(x_1 - x_2)^2 + (y_1 - y_2)^2} \right) \quad (5)$$

$$T = \begin{bmatrix} x'_1 \\ y'_1 \end{bmatrix} - R \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \quad (6)$$

식 (5)에서 얻은 θ 값은 식 (6)에서 회전행렬을 구하는데 쓰인다. 이렇게 얻은 R 과 T 를 이용하여 다른 쌍들에게도 적용한다. 정확한 R 과 T 를 구했다면 local map에서의 점을 변환하여 얻은 좌표값과 reference map에서의 좌표값이 일치할 것이다. 이를 점수화하기 위해 좌표변환으로 얻은 좌표와 짝지어진 좌표 간 차이의 제곱합을 오차로 하여 모든 쌍에서의 오차값들을 더한다. 이 때 나오는 오차들의 합을 점수로 정한다.

임의의 두 쌍을 골라 이 과정을 반복하여, 가장 낮은 점수값이 나오는 R, T 행렬을 최종 변환행렬의 구성요소로 정한다. 이것이 두 맵의 관계를 정의한다. 다음의 pseudo-code는 이 과정을 설명한다.

Transformation searching algorithm

```

FOR i = 1 to Number of elements of S
  FOR j = 1 to Number of elements of S
    get a pair  $P_i(x_1, y_1 \rightarrow x'_1, y'_1)$ 
    get a pair  $P_j(x_2, y_2 \rightarrow x'_2, y'_2)$ 
    find  $\theta$  satisfying  $P_i, P_j$  transformation
    get T with given  $\theta$ 
    score =  $\sum_k \left\| \left( R \begin{bmatrix} x_k \\ y_k \end{bmatrix} + T \right) - \begin{bmatrix} x'_k \\ y'_k \end{bmatrix} \right\|$ 
    find minimum score
  ENDFOR
ENDFOR

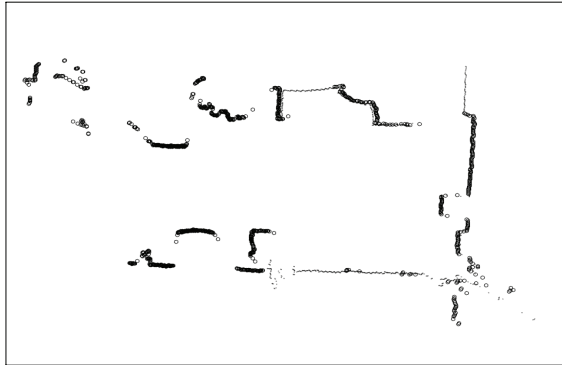
```

III. 실험

실험에 사용된 데이터는 단순한 실내 환경이 아닌 불특정 물체(책상, 의자, 서랍 등)가 위치한 일반 사무실 환경에서 LIDAR를 이용하여 수집하였다. 이 알고리즘의 목적은 두 개의 위치에서 찍은 point map 두 개를 일치시켜 센서가 어느 방향으로 얼마나 이동하였는지 파악하는 것이므로, 50개의 데이터를 얻어 서로 가까운 데이터끼리 짝을 지어 실험하였다. 그중에서도 ICP의 성능이 일정수준을 넘는 30개 쌍에 대해서만 실험을 진행하였다. 검출되는 모서리들이 90도가 아닌 것들도 다수 존재하여 최소값이 60도가 되도록 하였다. 이런 점들도 주변환경을 반영한 descriptor를 가지고 있기 때문에 좋은 위치추정에 충분히 기여한다. 그림 3의 (a)와 (b)는 서로 1m정도의 거리에 떨어진 거리에서 단일 LIDAR를 이용하여 데이터를 얻은 후 SPT로 위치추정을 한 모습이다. 그림에서 얇은 점으로 이루어진 data와 굵은 점으로 이루어진 data는 서로 다른 위치에서 얻은 data이다. (a)에서 다른 위치에서 찍은 데이터를 그대로 나타낸 것이



(a)



(b)

그림 3. (a) 서로 다른 위치에서 얻은 LIDAR data (b) SPT를 이용한 위치추정 결과.

Fig. 3. (a) LIDAR data taken from another location (b) Pose tracking result using SPT.

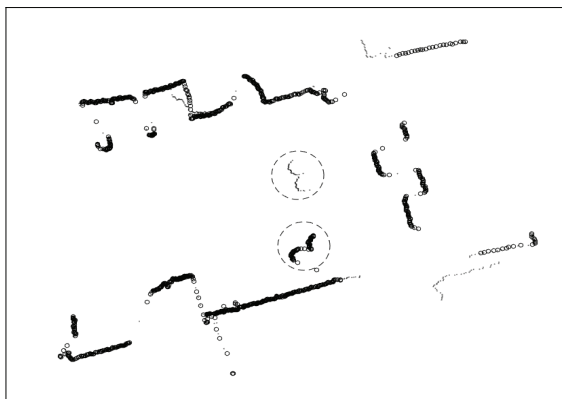


그림 4. 변화된 사람의 위치.

Fig. 4. Searching the right wall using RANSAC.

고, (b)는 알고리즘을 이용하여 transformation matrix를 구해서 두 데이터를 정확하게 일치시킨 그림이다.

주변 사물을 이동하여 지형을 변형시키거나, 사람이 지형을 가린 경우에도 이를 무시하고 정확히 자신의 위치를 찾는다. 그림 4는 사람의 위치가 변경된 경우에 SPT를 이용하여 위치추정을 한 예시이다. 그림에서 원으로 표시한 부분이 각각 사람의 다리가 LIDAR에 측정된 것이다. 얇은 점으로 표현된 데이터와 굵은 선으로 표현된 데이터가 방해물(사람의 다리)에도 정확히 두 데이터를 일치시키는 양

표 1. SPT와 ICP의 비교.

Table 1. Comparison between SPT and ICP.

	평균 변위 오차(m)	평균 회전 오차(rad)	평균 연산 시간(초)
SPT	0.022	0.0083	0.5439
ICP	0.637	0.048	0.3188

표 2. SPT와 ICP의 비교 예시.

Table 2. Examples of Comparison between SPT and ICP.

	서로 다른 위치에서의 LIDAR data	ICP 추정결과	SPT 추정결과
(1)			
(2)			
(3)			
(4)			

상을 볼 수 있다.

알고리즘의 성능을 정량적으로 나타내기 위해 수집하여 짝지는 데이터를 각각 ICP와 SPT에 사용하였다. 각 데이터 쌍에서의 translation과 orientation은 수작업으로 구하여 이를 ground truth로 사용하였다. 표 1은 이렇게 만든 ground truth와 측정값 사이의 오차를 ICP와 SPT에 대해 각각 구하여 비교한 것이다. 표 2는 이 실험에서 원본 데이터를 ICP와 SPT에 각각 적용하여 얻은 결과들 중 몇 개를 예시로 나타낸 것이다. 두 알고리즘 모두 이동정보(odometry)는 이용하지 않았으며, ICP의 경우 최대반복횟수를 1000회로 제한하였다. 1, 2번의 경우와 같이 두 알고리즘 모두 정확한 위치추정에 성공하는 경우도 있으나, 3, 4번의 경우 ICP는 odometry가 없어 국소 최저치에 수렴하는 모습을 보인다. 반면 SPT는 육안으로 잘 확인이 되지 않는 오차로 위치추정을 해냈다. 일정거리 이상 떨어진 거리에서 얻은 대부분의 data에서 ICP는 최적의 변환행렬을 구하는데 실패하였다. 또한 국소최저치에 빠질 경우 연산시간이 1초가 넘어가는 경우도 있었다.

IV. 결론

위치가 많이 떨어진 지점에서 찍은 지도는 서로 공유하는 점들이 많이 사라진다. 이에 따라 ICP는 오차가 많아지는 경향이 있다. 또한 국소 최저치에 수렴하면 연산속도가 현저하게 느려지기도 한다. 반면, SPT를 사용하면 모서리부분만을 특징점으로 사용하기 때문에 모서리들만 올바르게 짝지어지면 오차가 0에 가깝게 된다. 또한 모서리 개수에 정비례하여 속도가 증가하기 때문에 연산속도에 상한선이 존재한다.

SPT를 이용한 위치추정은 대체적으로 성공적이었고, 장애물이 존재하거나 지형이 변형이 된 경우에도 일부 모서리들의 정보만으로도 방해요소를 배제할 수 있었다. 다만, 모서리의 개수가 절대적으로 부족한 환경(복도, 넓은 면이 우세한 환경)에서는 잘 동작하지 못하였다. 따라서 후속 연구에서는 모서리뿐만 아니라 면에도 특징을 부여하여 위치추정을 하는 방법을 접목시킬 수 있을 것이다.

REFERENCES

- [1] J.-B. Song and S.-Y. Hwang, "Past and state-of-the-art SLAM technologies," *Journal of Institute of Control, Robotics and Systems (in Korean)*, vol. 20, no. 3, pp. 372-379, 2014.
- [2] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robots*, MIT press, 2011.
- [3] L. Jetto, S. Longhi, and D. Vitali, "Localization of a wheeled mobile robot by sensor data fusion based on a fuzzy logic adapted Kalman filter," *Control Engineering Practice*, vol. 7, no. 6, pp. 763-771, 1999.
- [4] J. Yang, J. Yang, and Z. Cai, "An efficient approach to pose tracking based on odometric error modelling for mobile robots," *Robotica*, vol. 33, no. 6, pp. 1231-1249, 2015.
- [5] B. Siemiątkowska and A. Zychewicz, "The application of ICP and sift algorithms for mobile robot localization," *ROMANSY 18 Robot Design, Dynamics and Control*, Springer Vienna, pp. 391-398, 2010.
- [6] F. Lu and E. Milios, "Robot pose estimation in unknown environments by matching 2D range scans," *Journal of Intelligent and Robotic Systems*, vol. 18, no. 3, pp. 249-275, 1997.
- [7] Graovac, Davor, Srećko Jurić-Kavelj, and Ivan Petrović, "Mobile robot pose tracking by correlation of laser range finder scans in Hough domain," *19th IEEE International Workshop on Robotics in Alpe-Adria-Danube Region*. 2010.
- [8] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte carlo localization for mobile robots," *Robotics and Automation, 1999. Proc. of the 1999 IEEE International Conference*, vol. 2. IEEE, 1999.
- [9] M. A. Fischler and R. C. Bolles, "Random sample

consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381-395, 1981.

- [10] S. Se, D. Lowe, and J. Little, "Global localization using distinctive visual features," *Proc. of Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference*, vol. 1. IEEE, 2002.
- [11] P. Bergström and O. Edlund, "Robust registration of point sets using iteratively reweighted least squares," *Computational Optimization and Applications*, vol. 58, no. 3, pp. 543-561, 2014.
- [12] H. Kim, D. Lee, T. Oh, and H. Myung, "Localization of a monocular camera using a feature-based probabilistic map," *Journal of Institute of Control, Robotics and Systems (in Korean)*, vol. 21, no. 4, pp. 367-371, 2015.



김 지 수

2015년 고려대학교 생체의공학과 학사. 2015년~현재 서울대학교 융합과학기술대학원 석박통합과정. 관심분야는 영상처리, 신호처리, 기계학습.



곽 노 준

1997년 서울대학교 전기공학부 학사. 1999년 서울대학교 전기공학부 석사. 2003년 동 대학 박사. 2003년~2006년 삼성전자 통신연구소 책임연구원. 2006년~2007년 서울대학교 BK21 정보기술사업단 BK 조교수. 2007년~2013년 아주대학교 전자공학부 조/부교수. 2013년~현재 서울대학교 융합과학기술부 부교수. 관심분야는 패턴인식, 기계학습, 영상처리, 컴퓨터비전.