

영상 교시기반 주행 알고리즘 성능 평가

Performance Evaluation of Visual Path Following Algorithm

최 이 삭, 하 중 은*
(I-Sak Choi¹ and Jong-Eun Ha¹)

¹Seoul National University of Science and Technology

Abstract: In this paper, we deal with performance evaluation of visual path following using 2D and 3D information. Visual path follow first teaches driving path by selecting milestone images then follows the same route by comparing the milestone image and current image. We follow the visual path following algorithm of [8] and [10]. In [8], a robot navigated with 2D image information only. But in [10], local 3D geometries are reconstructed between the milestone images in order to achieve fast feature prediction which allows the recovery from tracking failures. Experimental results including diverse indoor cases show performance of each algorithm.

Keywords: autonomous navigation, path follow, feature matching, intelligent robot

I. 서론

다양한 센서들이 로봇의 자율 주행을 위해 사용되고 있으며, 2007년 개최된 Urban Challenge 대회를 통해 도심 속의 일상적인 운전 환경과 유사한 환경에서 자율 주행이 가능함이 입증되었다[1]. 레이저 거리 센서를 이용한 실내 및 실외에서의 다양한 환경하에서의 자율 주행이 가능함이 보여졌다[2,3]. 이들 접근 방법에서 레이저 거리 센서가 주요 센서로 이용되고 있다. 레이저 거리 센서의 경우 다른 센서에 비해 정확한 정보를 제공하는 장점이 있으나 고비용과 처리 시간 단축의 어려움이 있다. 이러한 문제를 해결하기 위해 최근에는 상대적으로 저비용이면서 풍부한 정보를 제공하는 카메라를 이용한 자율 주행에 관한 연구가 활발히 진행되고 있다[4-7]. 한편, 카메라의 경우 정보 추출을 위한 추가적인 복잡한 처리 알고리즘이 필요하다는 단점이 있다.

본 연구에서는 먼저 주행 경로를 교시후 이를 실제 주행시 영상 정보를 이용하여 동일한 경로를 유지하는 방법론에 대해 다루도록 한다. 이러한 주행 기법은 영상 교시 기반 주행 [8-11]으로 불리고 있다. Chen과 Birchfield [8]는 이미지의 특징점의 추출 및 추적을 이용한 영상 교시 기반 주행 방법을 제안하고 있다. [9]에서는 로봇의 Odometry 정보를 추가로 이용하여 [8]의 방법을 확장하였다. [10,11]에서는 마일스톤 이미지 구축시 특징점의 삼차원 정보를 이용하고 있다. 삼차원 정보의 사용을 통해 보다 강인한 경로 추종 결과를 보이고 있다.

영상의 특징점 정보만을 이용한 영상 교시 기반 주행 방법 [8]에서는 교시단계와 상이한 주행환경에 대해서 특징 정보의 손실에 대한 문제를 발생시킨다. [10]에서는 [8]에서 거론된 문제점에 대한 강인한 알고리즘을 제시했다. [10]에서는

마일스톤 이미지 구축 시에 특징점의 삼차원 정보를 저장하고 이를 주행 단계에서 재투영 시킴으로써 주변 환경 변화와 갑작스런 장애물에 의한 특징점 손실에 대한 회복이 가능함을 보였다. 본 연구에서는 동일 주행 환경에 대해 이차원 [8] 및 삼차원 정보 [10]를 이용한 영상 교시 기반 주행 방법을 적용하여 성능을 분석하도록 한다.

II. 2D 영상 교시 기반 주행 알고리즘

본 장에서는 [10]의 기반이 되는 Chen&Birchfield [8]가 제안한 영상기반의 경로추종 알고리즘에 대해 살펴보고자 한다. 기본적인 과정은 교시 단계와 주행 단계로 구성된다 교시 단계에서는 수동으로 로봇을 조종하여 정해진 경로를 주행하면서 로봇에 장착되어 있는 카메라를 이용하여 이미지를 획득한다. 주행 중에 촬영된 이미지는 일정한 주행 시간 간격으로 분류된다. 분류된 각 단계의 시작 이미지 내에서 특징점을 추출하고 연속된 마지막 이미지까지 이를 추적하는 과정을 거친다. 여기서 각 단계의 시작 이미지는 마일스톤 이미지로 결정되고 각 특징점들의 초기 위치와 최종 위치 정보도 같이 저장한다.

주행 단계에서는 그림 1과 같이 교시 단계에서의 초기 출발 위치 근처에 로봇을 위치시킨다. 그리고 순차적으로 불러들인 마일스톤 이미지 내의 특징점들과 현재 이미지 내의 특징점들을 비교한다. 각각의 특징점 간의 상대적인 위치는 로봇의 진행방향을 결정하는 요소가 된다. 다음은 두 특징점들 간의 상관 관계에 따른 로봇의 조향 방향에 대한 판단식이다 [8].

$$\begin{cases} f_i^c > 0 \text{ and } f_i^d < 0 \text{ then turn right} \\ f_i^c > 0 \text{ and } f_i^c > f_i^d \text{ then turn right} \\ f_i^c < 0 \text{ and } f_i^d > 0 \text{ then turn left} \\ f_i^c < 0 \text{ and } f_i^c < f_i^d \text{ then turn left} \\ \text{otherwise do not turn} \end{cases} \quad (1)$$

* 책임저자(Corresponding Author)

논문접수: 2011. 4. 18., 수정: 2011. 7. 6., 채택확정: 2011. 7. 20.

최이삭: 서울과학기술대학교 NID융합기술대학원

(isakchoi@seoultech.ac.kr)

하중은: 서울과학기술대학교 자동차공학과(jeha@seoultech.ac.kr)

※ 본 논문은 2011 제26회 ICROS 학술대회에서 초안이 발표되었음.

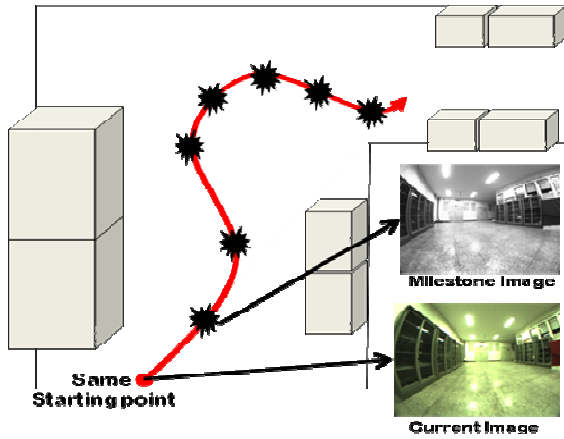


그림 1. 영상 교시 기반의 자율 주행.
Fig. 1. Autonomous navigation using visual path-follow.

f_i^c 는 현재 이미지상의 i 번째 특징점의 x 좌표값이며, f_i^d 는 마일스톤 이미지상의 대응하는 특징점의 x 좌표값이다. 영상상의 특징점 좌표는 이미지 중심을 원점으로 하여 계산한다. 이미지 중심의 오른쪽 부분의 x 좌표는 양수이며 왼쪽 부분은 음수이다. [8]에서 제안한 알고리즘은 위와 같이 추출된 특징점들의 x 좌표값만을 이용하여 영상 서보잉을 통해 목표지점으로 자율 주행한다. 전진 속도의 경우 일정 값을 유지하며 위와 같이 조향 방향에 대해서만 bang-bang 제어 기법을 통한 목표 지점에서의 도달을 수행하고 있다. 로봇은 순차적으로 해당 단계의 마일스톤 이미지를 통과함으로써 목표 지점에 도달하게 된다. 마일스톤 이미지의 특징점과 현재 이미지의 특징점이 서로 일치할 때 현재 단계는 완료되고 다음 단계의 마일스톤 이미지가 새로운 표적이 된다. 마일스톤 친이 조건은 특징점들 간의 x 좌표값의 평균 오차를 이용한다. 현재 특징점이 마일스톤 이미지의 특징점에 접근할수록 오차는 감소하며, 벗어날수록 오차는 증가하게 된다. [8]에서는 오차값이 감소하다가 증가하는 시점을 마일스톤 이미지 변경 시점으로 사용하였다.

III. 3D 영상 교시 기반 주행 알고리즘

본 장에서는 Diosi et al. [10]의 삼차원 정보를 이용한 영상 교시 기반주행 알고리즘에 대해 살펴보도록 한다. 교시 단계에서는 기존 [8]의 방법을 따라 시작 이미지 내에서 추출된 특징점은 마일스톤 종료 시기까지 연속된 이미지 내에서 추적하는 과정을 거친다.

특징점의 초기위치와 추적 위치의 상대적인 관계는 다음과 같이 표현된다.

$$q'^T F q = 0 \tag{2}$$

q' 는 특징점의 초기 위치를 가리키고 q 는 특징점의 다음 영상에서의 대응 위치이다. F 는 기본행렬(Fundamental Matrix)을 의미하고 카메라의 내부행렬과 카메라의 물리적인 자세 변환 정보를 포함한다. 카메라의 내부행렬을 아는 경우 식 (2)는 다음과 같이 변환할 수 있다.

$$p'^T E p = 0 \tag{3}$$

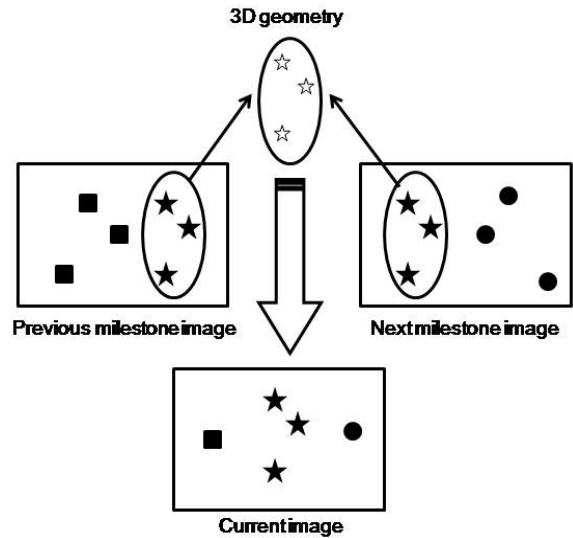


그림 2. 현재 이미지로 투영되는 마일스톤 특징점.
Fig. 2. Milestone features projected into current image.

E 는 필수 행렬(Essential Matrix)을 의미하고 p', p 는 내부 행렬에 의해 정규화된 좌표를 의미한다. 필수 행렬 E 를 구하기 위해 5 포인트 알고리즘 [13]을 이용한다. 5포인트 알고리즘은 서로 대응하는 5점 p', p 를 이용하여 필수 행렬을 구하는 알고리즘이다. 필수 행렬 E 를 구하기 위해 발생하는 오차를 최소화하기 위해 RANSAC [14] 알고리즘을 적용했다. 필수 행렬 E 가 결정되었다면 서로 대응하는 특징점 좌표를 이용하여 특징점의 삼차원 좌표를 구할 수 있다. 그리고 특징점의 삼차원 정보를 이미지에 재투영함으로써 삼차원 기하 정보의 오차율을 판단할 수 있게 된다. 앞서 언급한 과정은 연속된 이미지에서 반복 시행된다. 만약 삼차원 기하 정보의 오차율이 0.1(pixel) 이상을 넘기게 되면 마일스톤 이미지를 재지정한다. 이 때 특징점의 삼차원 좌표, 이차원 좌표, 마일스톤 시작 이미지를 이전 마일스톤 정보로 저장한다. 이후 연속된 이미지에서 특징점을 재추출하고 동일한 과정을 연속된 마지막 이미지까지 반복한다.

주행 단계에서는 시작지점에서 로봇은 현재 이미지와 가장 가까운 마일스톤의 특징점을 현재 이미지에 투영함으로써 처음 자세를 판단한다. 이후 실시간으로 획득된 이미지를 통해 특징점의 위치는 갱신되고 마일스톤 종료지점과의 상관관계를 이용하여 로봇의 조향 방향을 결정한다. 이와 동시에 마일스톤 특징점의 초기 위치 정보와 대응하는 현재 이미지의 특징점의 위치정보를 이용해서 카메라의 실시간 자세 변화를 추정한다. 카메라의 자세 변화는 교시 단계에서 언급한 RANSAC 알고리즘을 통한 5포인트 알고리즘으로 얻어낸 필수행렬로부터 얻어낸다. 실시간 이미지에 대한 필수 행렬을 이용하면 그림 2와 같이 교시단계에서 보존했던 삼차원 특징점을 현재 이미지에 투영시킬 수 있다. 주행단계에서 특징점의 투영은 마일스톤 당시에 추출했던 특징점의 삼차원 좌표와 주행 중의 카메라의 자세 정보를 이용하여 현재 이미지에 특징점을 투영하는 과정이다. 즉 현재 이미지 상에는 마일스톤에 의해 초기 위치가 결정되어 추적되고 있는 특징점과 현재 이미지의 카메라 자세 정보의 영향을 받아 투

영된 특징점이 공존한다. 그림 3은 성향이 서로 다른 두 특징점을 보여주고 있다. 빨강색 표식 'X'은 실시간으로 추적되고 있는 특징점이고, 검은색 표식 'O'은 투영된 특징점이다. 만약 현재 이미지를 통해 얻어낸 삼차원 기하 정보의 오차가 낮다면 성향이 다른 두 특징점들의 위치는 서로 일치할 것이다.

투영된 특징점은 다음과 같은 특성을 보인다. 만약 실세계 공간 상의 추적 대상이 되는 사물과 카메라 사이에 갑작스런 장애물이 등장했다고 가정할 때 현재 이미지에서는 추적 대상을 분별을 할 수 없다. 이 경우, 실시간으로 추적되는 특징점은 더 이상 추적이 불가능하기 때문에 제거된다. 하지만 재투영된 특징점은 카메라의 자세 정보에만 의존한 정보이기 때문에 분별이 되지 않는 추적 대상의 예상 위치를 예측할 수 있다. 그림 4에서 왼쪽 그림은 일반적으로 실세계 공간상의 대상을 추적하는 특징점과 마일스톤 특징점의 삼차원 좌표에 의해 재투영된 특징점을 보여준다. 오른쪽 그림은 실세계 공간상의 사물과 실제 카메라 사이에 장애물이 존재할 경우 특징점의 삼차원 좌표만이 현재 이미지에 투영될 수 있음을 보인다.

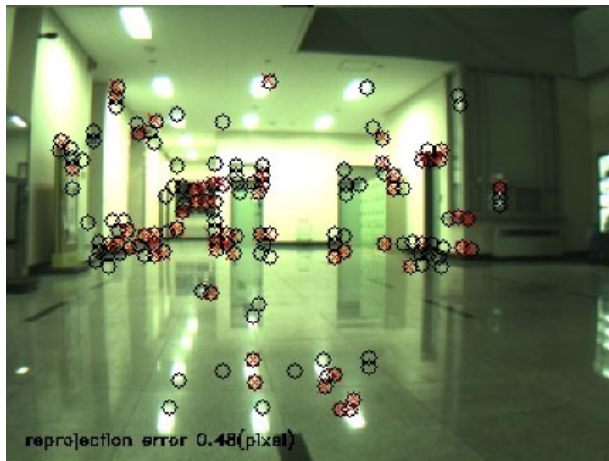


그림 3. 주행 결과 이미지 (빨강색 'X': 추적되는 특징점, 검은색 'O': 투영되는 특징점).
Fig. 3. Result image of navigation (red 'X': tracked feature point, black 'O': projected feature point).

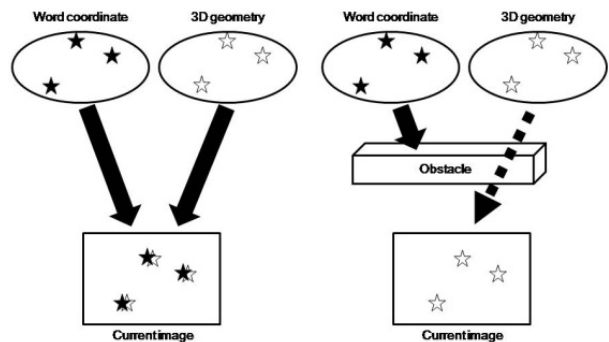


그림 4. 투영된 특징점의 특성 (왼쪽: 일반적인 경우, 오른쪽: 장애물이 발생할 경우).
Fig. 4. Characteristics of projected feature point (left: general case, right: obstacle case).

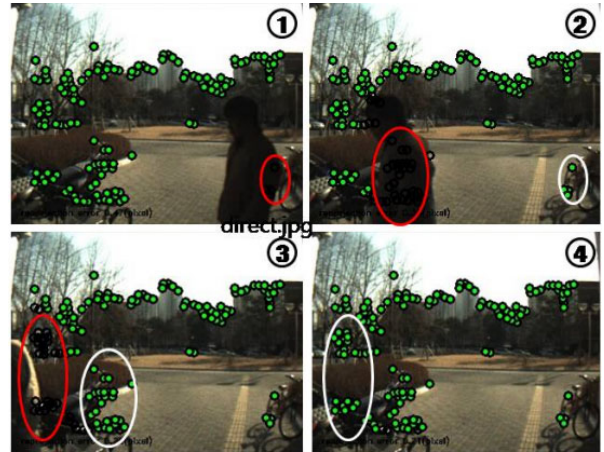


그림 5. 장애물에 의해 손실된 특징점의 회복.
Fig. 5. The recovery of feature point that has been lost by obstacle.

주변 환경 변화나 갑작스런 장애물에 의해 추적되지 못하고 제거된 특징점은 투영된 특징점에 의해 복구될 수 있다. 특징점 복구 과정은 다음과 같다. 마일스톤 변환 시기에 검출에 성공한 특징점 좌표의 이미지 패치를 KLT tracker [12]의 방식에 따라 저장한다. 그리고 주행 단계에서 주변 환경 변화의 영향으로 추적중인 특징점이 사라지게 되면 투영 좌표 주변의 이미지와 마일스톤 변환 시기에 저장했던 이미지 패치 정보를 실시간으로 비교한다. 그리고 장애 요인이 사라지게 되어 비교된 정보가 유사할 경우 사라진 특징점은 다시 복구된다. 그림 5는 실제 주행 단계에서 갑작스런 외부의 장애로 인해 특징점이 사라지고 이 후 특징점의 복구 되는 과정을 보여준다. 녹색 원은 실시간으로 추적되고 있는 특징점, 검은 원은 재투영된 특징점, 빨강 원은 사라진 특징점, 흰색 원은 회복된 특징점을 의미한다. 보행자로 보이는 장애물이 오른쪽에서 왼쪽으로 카메라 앞을 이동함에 따라 실시간으로 추적되는 특징점은 사라지지만 재투영된 특징점은 여전히 남아있다. 이 후 장애물이 시야에서 사라지면 실시간으로 추적되고 있는 특징점이 다시 복구되는 것을 확인할 수 있다.

5포인트 알고리즘을 통해 얻어낸 필수 행렬 E 는 고정된 스케일 값을 가지지 않는다. 교시단계에서 획득한 특징점의 삼차원 좌표를 주행 단계에서 현재 이미지에 투영되기 위해서는 각각의 필수 행렬의 스케일 값을 일치시켜야 한다. 마일스톤의 특징점과 주행단계에서의 특징점은 세계좌표에 존재하는 동일한 사물의 위치 정보를 이미지 좌표로 투영한 점 이므로 두 경우의 삼차원 좌표 정보를 이용해서 상대적인 스케일 정보를 구할 수 있다. 서로 대응하는 마일스톤 특징점의 삼차원 좌표와 현재 이미지를 이용하여 획득한 삼차원 좌표의 스칼라 비율을 이용하여 스케일 값을 일치시킨다.

IV. 실험 결과

실험은 MoileRobot사의 P3-AT 이동로봇에 노트북(Dell studio 14)을 장착하여 수행하였다. 카메라는 어안렌즈를 장착한 Point Grey사의 FireFly를 이용하였으며 그림 6과 같이 로봇의 전진방향과 평행하게 설치하였다. 로봇의 주행 궤적은 P3-AT로봇에서 제공하는 자세 정보와 종료지점의 수작업에 의한 실측 정보를 이용하였다. P3-AT의 자세 정보는 엔코더



그림 6. 실험에 사용한 로봇 (MobileRobot's P3-AT).
Fig. 6. Robot used in experiments (MobileRobot's P3-AT).

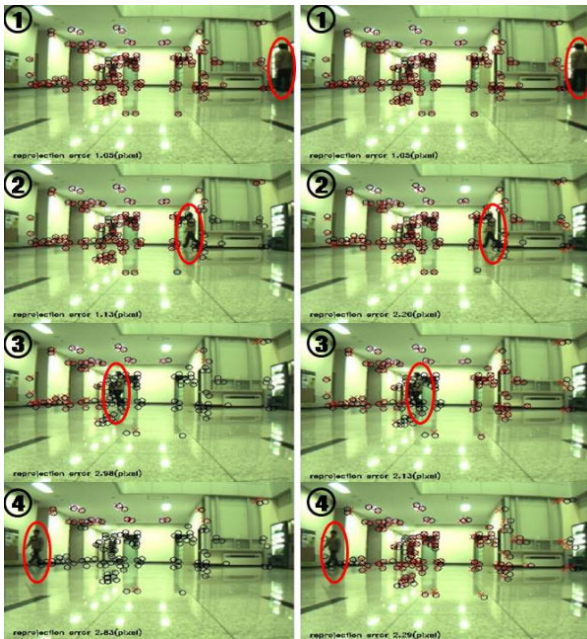


그림 7. 각 알고리즘에 의한 특징점 회복 결과 이미지 (왼쪽: [8]의 알고리즘, 오른쪽: [10]의 알고리즘. 빨강색 'X': 추적되는 특징점, 검은색 'O': 투영되는 특징점, 빨강색 원: 장애물).
Fig. 7. Result of feature point recovery according to algorithm (left: [8]'s algorithm, right: [10]'s algorithm. red 'X': tracked featured point, black 'O': projected feature point, red circle: obstacle).

및 자이로 센서를 이용하고 있으며 바퀴의 슬립이 발생하는 회전 이동에서는 부정확한 위치 정보를 제공한다. 따라서 실제 측정을 통해 이를 보완하도록 한다.

실험 1에서는 가상 장애물을 발생시켜 삼차원 정보를 이용한 주행 알고리즘의 성능을 평가하도록 한다. 실험 방법은 12개의 실험 장소를 선정하여 주행 중인 로봇을 정지한 후에 보행자를 카메라 시야에 등장시켜 강제로 특징점의 손실을 가져오도록 하고 회복되는 정도를 판단하도록 한다.

그림 7의 왼쪽 부분은 특징점의 이차원 정보만을 이용한 실험이고 오른쪽 그림은 특징점의 삼차원 정보를 이용하여 특징점을 회복한 실험이다. 빨강색 표식 'X'은 실시간으로

표 1. 실험 1에서 특징점의 손실 및 회복 개수.

Table 1. The number of loss and recovery of feature point in experiment 1.

장소	손실 (개수)	회복 (개수)	회복율(%)
1	85	68	80.0
2	16	15	93.7
3	9	8	88.8
4	16	10	62.5
5	37	21	56.7
6	11	11	100
7	42	42	100
8	49	42	85.7
9	9	8	88.9
10	115	102	88.7
11	107	106	99.1
12	109	98	89.9

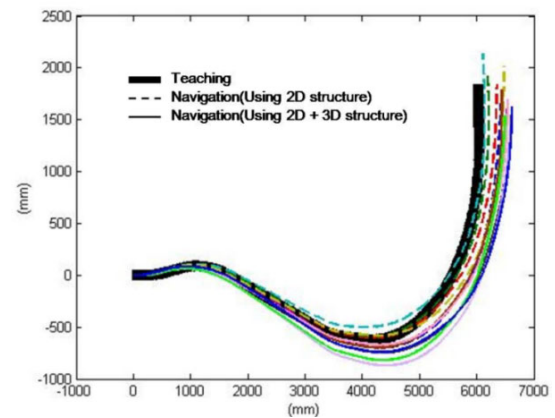


그림 8. 실험 2의 주행 궤적 비교 (검정색: 교시단계, 점선: [8], 실선: [10]).

Fig. 8. Comparison of running locus in experiment 2 (black line: teaching phase, dot line: [8], solid line: [10]).

추적되고 있는 특징점, 검은색 표식 'O'은 채투영된 특징점, 빨강색 타원은 장애물을 의미한다. 왼쪽 그림에서는 보행자가 카메라 시야 내에서 이동함에 따라 현재 이미지에 추적되는 특징점의 손실만 되는 반면 오른쪽 그림에서는 특징점의 손실과 회복을 볼 수 있다. 표 1은 특징점의 삼차원 정보를 이용했을 때 장애물에 의해 손실되는 특징점에 대한 회복 성능을 보여준다. 총 12회에 걸친 테스트에서 전체 손실에 대한 회복율은 평균 86.18%의 결과를 보인다.

실험 2는 특징점의 이차원 정보만을 이용한 알고리즘 [8]과 삼차원 정보를 이용하는 알고리즘 [10]을 이용한 동일 환경에서의 주행 결과를 통해 성능을 분석한다. 분석 방법은 교시단계와 주행 단계의 주행 오차를 평가하고 특징점의 회복 성능을 판단한다. 연구실 내에서 각 알고리즘 방법을 이용하여 6회 주행을 시행하였으며 교시 단계에서 획득한 2291 장의 이미지에서 122 장의 마일스톤 이미지가 사용되었다. 로봇의 교시단계 속도는 선속도 100 mm/s, 각속도 5 deg/s를 이용하고 주행 단계에서는 선속도 50 mm/s, 각속도 4 deg/s를 이용하였다. 그림 8은 교시 단계의 정보를 이용하여 주행한 로봇의 궤적이다. 검은색 굵은 선은 교시 단계에

표 2. 실험 2의 결과.

Table 2. Result of experiment 2.

교시	주행 종료지점		특징점 손실(개)	특징점 회복(개)
	P3-AT 좌표 [x, y] /오차(mm)	실측 오차 (mm)		
교시	[6031,1836]			
2D	1	[6358,1841] /327	122	1138
	2	[6535,1626] /549	121	802
	3	[6429,1658] /435	94	1093
	4	[6195,1957] /203	143	1023
	5	[6109,2140] /313	133	1210
	6	[6479,2013] /481	282	839
2D + 3D	1	[6445,1788] /416	185	792
	2	[6616,1563] /645	173	653
	3	[6547,1700] /533	174	658
	4	[6590,1398] /710	134	580
	5	[6506,1543] /557	135	572
	6	[6621,1627] /625	221	850

표 3. 실험 3의 결과.

Table 3. Result of experiment 3.

교시	주행 종료지점		특징점 손실(개)	특징점 회복(개)
	P3-AT 좌표 [x, y] /오차(mm)	실측 오차 (mm)		
교시	[20327,-3100]			
2D	1	[20853,-1966] /1169	72	695
	2	[21041,-2658] /796	101	620
	3	[20801,-2307] /847	134	856
	4	[20935,-1434] /1689	95	1154
	5	[20405,-1722] /1290	631	921
	6	[21287,-1083] /2193	222	899
2D + 3D	1	[20803,-2526] /678	72	887
	2	[21169,-1401] /1815	2	909
	3	[21066,-864] /2264	44	731
	4	[21372,-1030] /2238	91	884
	5	[21348,-1621] /1723	102	948
	6	[21345,-2057] /1394	133	1119

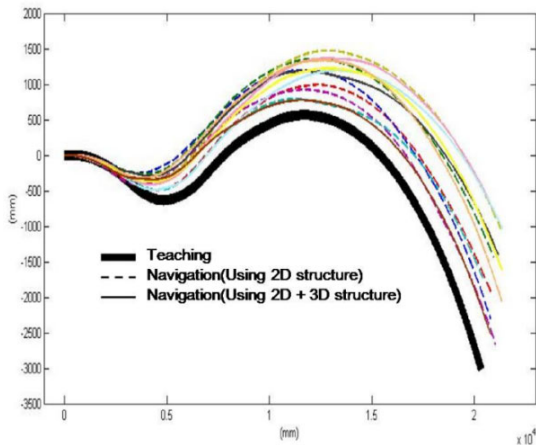


그림 9. 실험 3의 주행 궤적 비교 (검정색: 교시단계, 점선: [8], 실선: [10]).

Fig. 9. Comparison of running locus in experiment 3 (black line: teaching phase, dot line: [8], solid line: [10]).

서의 로봇의 주행 궤적이고 점선은 주행 단계에서 [8]을 이용한 로봇의 주행궤적을 의미하며 실선은 [10]을 이용한 로봇의 주행궤적이다.

표 2는 총 12회에 걸친 주행 실험에서 [8]과 [10]의 주행 오차와 특징점의 회복 정도를 보여주고 있다. [8]의 경우 P3-AT 평균 주행 오차와 실측 오차는 각 384.7 mm, 146 mm, [10]의 경우 P3-AT 평균 주행 오차와 실측 오차는 각 581.5 mm, 167.5 mm의 오차를 보였다. 주행 종료 지점을 통한 주행 오차를 보면 두 알고리즘은 큰 성능의 차이를 보이지 않는다. 하지만 특징점의 손실 여부를 판단할 경우 [8]은 평균 1017개의 특징점을 손실했고 [10]은 684개의 특징점을 손실하고 146개의 특징점이 회복되었다.

실험 3은 실험 2에 비해 보다 넓은 건물 1층 로비에서 6회 시행되었다. 교시 단계에서 획득한 2100장의 이미지에서 154장의 마일스톤 이미지가 사용되었다. 로봇의 교시단계 속도는 선속도 300 mm/s, 각속도 5 deg/s를 이용하고 주행 단계에서는 선속도 150 mm/s, 각속도 4 deg/s를 이용하였다. 그림 9

는 교시 단계의 정보를 이용하여 주행한 로봇의 궤적이다. 검은색 굵은 선은 교시 단계에서의 로봇의 주행 궤적이고 점선은 주행 단계에서 [8]을 이용한 로봇의 주행궤적을 의미하며 실선은 [10]을 이용한 로봇의 주행궤적이다.

표 3은 총 12회에 걸친 주행 실험에서 [8]과 [10]의 주행 오차와 특징점의 회복 정도를 보인다. [8]의 경우 P3-AT 평균 주행 오차와 실측 오차는 각 1330.9 mm, 209.2 mm, [10]의 경우 P3-AT 평균 주행 오차와 실측 오차는 각 1686.9 mm, 74 mm의 오차를 보였다. 주행 종료 지점을 통한 주행 오차를 보면 두 알고리즘은 큰 성능의 차이를 보이지 않는다. 하지만 특징점의 손실 여부를 판단할 경우 [8]은 평균 857개의 특징점을 손실했고 [10]은 913개의 특징점을 손실하고 182개의 특징점이 회복되었다.

V. 결론

본 논문에서는 이미지상의 특징점들의 이차원 좌표만을 이용한 영상 교시 기반 알고리즘 [8]과 특징점의 삼차원 정보를 추가적으로 이용한 알고리즘 [10]의 성능을 동일한 주행 환경에서 비교 분석하였다. 두 알고리즘은 특징점의 이차원 좌표를 이용한 주행 방법을 사용하기 때문에 주행성 부분에서 거의 동일한 성능을 보인다. 하지만 주행 과정 중에 외부 환경 변화의 영향으로 특징점의 손실이 생길 경우 삼차원 정보를 이용하는 알고리즘은 특징점을 회복함으로써 보다 안정적인 성능을 보였다.

참고문헌

- [1] C. Urmson, "Autonomous driving in urban environments: boss and the urban challenge," *Journal of Field Robotics*, vol. 25, no. 8, pp. 425-466, 2008.
- [2] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," *Proc. of the AAAI National Conference on Artificial Intelligence*, 2002.
- [3] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, MIT Press, 2006.
- [4] R. Sim, P. Elinas, M. Griffin, A. Shyr, and J. J. Little, "Design

- and analysis of a framework for real-time vision-based SLAM using Rao-Blackwellised particle filters," *Proceedings of the 3rd Canadian Conference on Computer and Robot Vision (CRV'06)*, 2006.
- [5] S. Se, D. Lowe, and J. Little, "Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks," *The International Journal of Robotics Research*, vol. 21, no. 8, pp. 735-758, 2002.
- [6] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 1052-1067, 2007.
- [7] N. Karlsson, E. D. Bernardo, J. Ostrowski, L. Goncalves, P. Pirjanian, and M. E. Munich, "The vSLAM algorithm for robust localization and mapping," *IEEE International Conference on Robotics and Automation*, pp. 24-29, 2005.
- [8] Z. Chen and S. T. Birchfield, "Qualitative vision based path following," *IEEE International Conference on Robotics and Automation*, pp. 2686-2692, 2006.
- [9] Z. Chen and S. T. Birchfield, "Qualitative vision-based path following," *IEEE Transactions on Robotics*, vol. 25, no. 3, pp. 749-754, 2009.
- [10] A. Diosi, A. Remazeilles, S. Segvic, and F. Chaumette, "Outdoor visual path following experiments," *IEEE/RSJ International Conference on Robots and Systems*, pp. 4265-4270, 2007.
- [11] E. Royer, M. Lhuillier, M. Dhome, and J.-M. Lavest, "Monocular vision for mobile robot localization and autonomous navigation," *International Journal of Computer Vision*, vol. 74, no. 3, pp. 237-260, 2007.
- [12] S. Birchfield, "KLT: An implementation of the Kanade Lucas-Tomasi feature tracker;" [Online]. Available: <http://www.ces.clemson.edu/~stb/klf/>
- [13] D. Nister. "An efficient solution to the five-point relative pose problem," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 756-770, 2004.
- [14] M. Fischler and R. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Readings in computer vision: issues, problems, principles, and paradigms*, pp. 726-740, 1987.
- [15] I. S. Choi and J. E. Ha, "Performance comparison of visual path following using 2D and 3D information," *Proc. of 2011 26th ICROS Annual Conference (in Korean)*, Gwangju, Korea, pp. 275-280, 2011.



최 이 삭

2010년 서울과학기술대학교 자동차공학과 졸업. 2010년~현재 서울과학기술대학교 NID 융합기술대학원 재학중. 관심분야는 지능형 자동차/로봇, 무인 자율주행.



하 종 은

1992년 서울대학교 기계공학과 졸업. 1994년 서울대학교 기계공학과 석사. 2000년 한국과학기술원 기계공학과 박사. 2000년~2002년 삼성코닝. 2002년~2005년 동명대학교 멀티미디어공학과. 2005년~현재 서울과학기술대학교 자동차공학과. 관심분야는 지능형 자동차/로봇, 무인 자율 주행, 삼차원 복원, 머신 비전.