



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

공학석사학위논문

자율 주차를 위한 Around View Monitor(AVM) 기반 Visual SLAM

**Around View Monitor(AVM) Based Visual
SLAM For Autonomous Parking**

2021년 2월

서울대학교 융합과학기술대학원
지능정보융합학과 지능정보융합학전공
안찬우

**자율 주차를 위한 Around View
Monitor(AVM) 기반 Visual SLAM**
**Around View Monitor(AVM) Based Visual
SLAM For Autonomous Parking**

지도교수 박재홍

이 논문을 공학석사 학위논문으로 제출함
2020년 11월

서울대학교 융합과학기술대학원
지능정보융합학과 지능정보융합학전공
안찬우

안찬우의 석사 학위논문을 인준함
2020년 12월

위원장 곽노준

부위원장 박재홍

위원 전동석



초 록

자율주차는 인식, 계획, 제어로 이루어져 있다. 자율주차의 인식과정 동안에 자동차는 자신의 위치를 알아야하고 주변 환경을 인지할 수 있어야 한다. 이러한 과정을 동시적 위치추정 및 지도 생성(SLAM)이라고 한다. 많은 SLAM 알고리즘들은 주변 환경의 정확한 인식을 위해 제안되어왔다. 특히 값싼 카메라를 주된 센서로 사용하는 Visual SLAM은 자율주행자동차를 위한 유망한 알고리즘으로 여겨져왔다. 대부분의 Visual SLAM은 전방 카메라를 사용한다. 이러한 시스템 형태에서, Visual SLAM은 대부분의 환경에서 잘 작동한다. 그러나 주변환경에 특징점이 적고, 강한 빛이 있는 환경에서는 Visual SLAM의 성능이 하락한다. 많은 수의 주차장들은 야외에 위치해있고, 주차선과 같은 단조로운 특징점들만을 가지고 있다. 주차장에서의 이러한 문제들에 대처하고 Visual SLAM의 성능을 개선하기 위해서 이 연구에서는 Around View Monitor(AVM)을 주요 센서로 하는 새로운 Visual SLAM 알고리즘을 제안한다. AVM 시스템에서는 Top View 이미지가 생성되기 때문에 푸리에 변환은 AVM 이미지들로부터 동작 정보를 추출하기 위해서 사용된다. 동작정보를 추정하기 위해 reprojection error 또는 photometric error 등을 비용 함수로써 사용하는 기존의 Visual SLAM[1], [2], [3]과는 달리 푸리에 변환은 어떠한 특징점 매칭이나 최적화 과정없이, 참조되는 이미지로부터 대상이 되는 이미지로의 동작 정보를 간단히 추정할 수 있다. 또한 자동차의 위치를 정확하게 그리고 강건하게 추정하기 위해서 landmark를 이용한 위치추정방법이 사용되었다. 이 연구에서 landmark라함은 주차선끼리 만나는 Cross point(교차점)을 말한다. 이 연구에서 landmark를 이용한 위치추정은 세가지 단계로 나뉘

다. 첫번째 단계는 교차점을 탐색하는 것이다. 주로 이미지에서 특징점을 찾기 위해 Image Segmentation 방법을 사용하는 기존의 AVM 기반의 SLAM연구[4], [5]와는 달리 이 연구에서는 Image Segmentation 보다 훈련이 더 쉽고 간단한 Object Detection 네트워크인 YoloV3[6]를 사용하여 교차점을 탐색하였다. 두번째 단계는 Data Association이다. SLAM에서 Data Association은 지도에 등록되어 있는 특징점과 현재 관찰된 특징점을 서로 연관시키는 작업이다. Deep SORT[7]가 현재 관찰된 특징점을 추적하기 위해 사용되었지만, Deep SORT[7]를 사용할 때에는 추적되는 특징점의 ID가 비교적 자주 바뀌는 현상이 일어나서 푸리에 변환으로부터의 동작 정보와 현재 관찰된 특징점 정보를 사용하여 Nearest Neighbor[8] 방법을 통해 추가적인 Data Association을 구현하였다. 푸리에 변환으로부터의 동작 정보는 비교적 정확하고, 교차점 사이의 거리는 멀기 때문에 Data Association의 정확도는 Deep SORT[7]만 사용했을 때보다 정확해졌다. 이후에 마지막 단계에서는 일정 개수이상의 data association이 이루어졌을때, Singular Value Decomposition을 이용하여 새롭게 동작 정보가 추정된다. 기존의 Visual SLAM 과 제안된 SLAM 알고리즘의 성능을 비교하기 위해서 주차장에서 실험을 진행하였고, LOAM[9]이 알고리즘의 비교를 위한 Groundtruth로서 사용되었다.

주요어 : 동시적 위치추정 및 지도생성, 자율 주차, Landmark 기반 위치 추정, 푸리에 변환, Around View Monitor(AVM), Data Association

학번 : 2019-21330

목차

그림 목차	v
표 목차	vi
I. Visual Simultaneous Localization And Mapping(Visual SLAM)	1
1.1 Visual SLAM이란?	1
1.2 특징점 탐색	2
1.3 Data Association	3
1.4 Tracking	4
1.5 성능 평가	5
II. 자율 주차(Autonomous Parking)	7
2.1 자율 주차란?	7
2.2 자율 주차의 발전	7
2.3 상용 자율 주차	8
III. 자율 주차를 위한 AVM 기반 Visual SLAM	9
3.1 서론	9
3.2 방법	13
3.2.1 제안하는 SLAM 파이프라인	13
3.2.2 교차점의 탐색과 추적 그리고 모서리 추출	16
3.2.3 푸리에 변환을 이용한 위치 추정	18
3.2.4 Keyframe 생성	20

3.2.5	Data Association	22
3.2.6	교차점을 이용한 Landmark 기반의 위치 추정	24
3.3	실험	25
3.3.1	실험 준비	25
3.3.2	실험 결과	28
3.4	고찰 및 결론	33
3.4.1	고찰	33
3.4.2	결론	34
	참고 문헌	36
	Abstract	42

그림 목차

그림 1.	제안하는 SLAM 알고리즘의 전체 파이프라인	15
그림 2.	교차점 탐색과 추적 결과	17
그림 3.	AVM 이미지로부터의 모서리 추출 결과	17
그림 4.	실험에 사용한 자율주행용 자동차 시스템	27
그림 5.	실험을 진행한 주차장, 융합과학기술대학원 민원인 주차장	27
그림 6.	실험 결과의 그래프에 쓰이는 표시의 의미	28
그림 7.	Keyframe을 사용하지 않고 경로를 추정하였을 때의 결과. 각 그래프에서의 빨간색 타원은 전체 경로에 서 회전 오차가 눈에 띄게 개선이 된 부분을 표시한 것이다.	28
그림 8.	Keyframe을 사용하여 경로를 추정하였을 때의 결과 .	29
그림 9.	동작정보를 이용하여 Data Association을 하지 않을 때의 경로	29
그림 10.	동작정보를 이용하여 Data Association을 하였을 때 의 경로	30

표 목차

표 1.	교차점당 평균 몇 개의 아이디어가 할당되는지 계산한 결과	30
표 2.	주차 1 결과(Total length: 29.122m)	31
표 3.	주차 2 결과(Total length: 11.727m)	32

제 1 장

Visual Simultaeneous Localization And Mapping(Visual SLAM)

1.1 Visual SLAM이란?

Simultaeneous Localization And Mapping(SLAM)이란 지도의 생성과 위치 추정을 동시에 하는 것을 의미한다. 만약 센서를 통하여 주변 환경을 관찰하여 얻은 데이터가 자신이 현재 가지고 있는 지도에서 어떤곳에 대응되는지 알 수 있다면, 자신의 위치를 추정할 수 있다. 반대로 자신의 위치를 알고 있으면 그 데이터를 이용하여 현재 지도를 갱신할 수 있다. SLAM 기술은 센서를 장착하고 있는 모바일 로봇에서 주로 사용되며, 로봇의 안정적인 구동을 위해서 필수적인 기술이다. SLAM 알고리즘에는 대표적으로 광원을 이용하여 주변 물체를 인식하는 센서인 Light Detection And Ranging(LiDAR)를 사용하는 LiDAR SLAM, RGB-D 카메라를 사용하는 RGB-D SLAM, IMU를 사용하는 Visual Inertial SLAM 혹은 Visual Inertial Odometry. 그리고 마지막으로 RGB 카메라를 사용하는 Visual SLAM이 있다. Visual SLAM은 카메라로부터 얻은 이미지를 가지고 로봇의 위치를 파악하고 주변 지도를 생성하는 알고리즘이다. Visual SLAM의 전체 과정은 크게 보면 3가지 단계로 나눌 수 있다. 첫번째는 이미지에서 알고리즘에 사용할 특징점을 정의해야한다. 그리고 정의된 특징점을 이미지에서 어떠한 방식으로 찾을 것인지 생각해야한다. 이러한 과정을 특징점 탐색

이라고 한다. 두번째로 현재 이미지에서 찾은 특징점이 지도에 등록된 특징점들중에 어떤 특징점에 해당하는지 계산하는 Data Association 과정이 있다. 특징점간에 Data Association이 결정되었으면 마지막으로 Tracking 과정을 거쳐서 로봇의 움직임을 추정한다. 추정된 움직임 정보를 이용하여 앞서 찾은 특징점을 전체 지도에 추가한다. 이 과정에서 기존에 지도에 없던 특징점들은 새로이 지도에 추가되게 된다. 각 단계에는 SLAM 방법마다 다양한 방법들이 존재하는데, 자세한 내용은 다음 절들에서 자세히 다루겠다.

1.2 특징점 탐색

특징점 탐색에서 특징점은 SLAM 시스템에서 지도를 구성하는 요소들을 의미한다. 특징점은 지도에 등록됨과 동시에 현재 찾은 특징점을 이용하여 현재 지도에서 비슷한 특징점을 탐색함으로써 로봇의 현재 위치를 추정하는 것을 가능케 해주는 요소이다. 특징점은 SLAM 시스템을 구현하는 사용자에 의해, SLAM 시스템이 사용되는 환경에 따라서 다양하게 정의된다. Visual SLAM에서 대표적으로 사용되는 특징점으로는 RGB 카메라의 이미지에서 추출될 수 있는, 특징 기술자(feature descriptor)에 의해 정의되는 특징점이 있다. 이것은 이미지안에서 다른 부분에 비해 픽셀 변화가 눈에 띄는 코너점을 특징점으로서 탐색하고, 이미지안에서 특징점들끼리 특징 기술자에 의해 서로 구분되어 존재한다. 특징 기술자기반의 특징점으로는 ORB[10] 등의 다른 기술자에 비해 비교적 속도가 빠른 이진 기술자가 대표적이다. ORB[10]는 FAST[11] 특징점과 BRIEF[12] 특징 기술자가 합쳐진 것인데, FAST[11]의 빠른 특징점 탐색과 BRIEF[12]의 빠른 특징 기술 속도로 인해 ORB SLAM[1]과 같은 Visual SLAM 알고

리즘에도 쓰인 바가 있다. 이외에도 로봇이 움직이는 곳의 공간적 특성을 이용하여 정의된 특징점도 존재한다. 대표적으로 StructSLAM[13]에서는 로봇이 인공적인 건물에서 돌아다닌다는 가정을 한 후, 건물 구조의 선 정보를 특징점으로서 사용하였다. 이와 같이, 로봇이 움직이는 공간의 특성에 따라서 다양한 특징점들이 정의될 수 있고, 정의된 특징점이 탐색되는 방식은 정의된 특징점에 따라 매우 다양하다.

1.3 Data Association

특징점의 정의와 탐색이 끝났다면 그 뒤에 필수적으로 행해야 할 것은 Data Association이다. Data Association이란 현재 탐색된 특징점이 현재 지도에 등록된 특징점중 어떤 특징점에 해당하는지 판단하는 과정을 말한다. Visual SLAM에서 특징점 기반의 Data Association중 가장 대표적인 방법은 Nearest Neighbor(NN)[8], Multi-Hypothesis Tracker(MHT)[14], Joint Compatibility Branch and Bound(JCBB)[15], Simple Online and Realtime Tracking(SORT)[16]등의 방법이 있다. NN[8]은 현재 탐색된 특징점에 대해서 지도상에 가장 가까운 feature를 association시키는 방법이다. 지도상에서 가깝다는 것은 여러가지 metric이 있을 수 있는데, 측정되는 데이터의 종류에 따라서 유클리디언 거리, 마할라노비스 거리등 다양한 metric이 사용된다. MHT[14]는 탐색된 특징점 각각에 대해서 여러가지 가설을 보유하고 있으므로 Data Association의 성공률을 높인다. 하지만 특징점이 많아지고 가설의 개수가 많아짐에 따라서 계산 비용이 기하급수적으로 높아진다는 단점이 있다. 또한 JCBB[15]는 현재 탐색된 특징점과 지도상의 특징점간의 결합 분포 확률을 비교함으로써 Data Association의 성공률이 높지만, 특징점이 많아질 수록 계산 비용이 많다는 단점이 있다.

SORT[16]는 계산 비용이 높은 JCBB[15]의 대안책으로서 제시되었고, 선형 모델의 칼만 필터와 헝가리안 알고리즘을 결합한 알고리즘을 제시하였다. 이는 간단한 알고리즘 구조로도 JCBB[15] 이상의 성능을 보였다. 하지만 특징점이 선형 속도 모델을 따른다는 가정을 하였다는데에 한계점이 있다. 또한 최근에는 Convolutional Neural Network(CNN)을 이용한 Data Association 알고리즘도 각광을 받고 있다. 이러한 알고리즘에는 대표적으로 SORT[16]로부터 파생된 Deep SORT[7]가 있는데, 이는 CNN, 칼만 필터, 헝가리안 알고리즘의 결합으로 뛰어난 성능을 보인다.

1.4 Tracking

Visual SLAM에서 Tracking이라 함은 앞서 얻은 특징점의 Data Association 정보를 가지고 로봇의 위치 변화를 추정하는 것을 의미한다. 일반적으로 3차원 SLAM인 ORB SLAM[10] 이나 DSO[2]와 같은 알고리즘들은 비용함수를 정의한다. 전자는 reprojection error라는 개념을 도입하여 association된 특징점들을 3차원 공간에서 2차원 이미지 공간으로 재투영(reprojection)한 뒤에, 각 특징점의 이미지 좌표계에서의 위치 차이를 계산한것을 비용으로 정의를 하고, 후자는 photometric error라는 개념을 도입하여 위치 차이 대신에 픽셀 강도의 차이를 비용으로서 정의를 한다. 그 이후에는 각각 정의된 비용함수를 최적화하는 방식으로 로봇의 위치 변화를 찾는다. 그 외에 2차원 SLAM에서는 Iterative Closest Point(ICP) 계열의 알고리즘이 일반적으로 사용될 수 있다. 이는 점군(pointcloud)의 위치의 차이를 이용한 최소제곱 비용함수를 최적화하는 방식으로 로봇의 위치 변화를 찾는다.

1.5 성능 평가

Visual SLAM의 성능이 다른 알고리즘에 비해 얼마나 성능이 좋고 나쁜지 판단하기 위해서는 평가의 척도가 필요할 것이다. 어떤 trajectory에 대한 groundtruth를 X_{gt} , estimated trajectory를 X , estimated trajectory가 groundtruth에 대해 align된 trajectory를 \hat{X} 라고 할 때, 두가지의 성능 비교 척도가 존재한다. 첫번째는 Absolute Trajectory Error(ATE), 두 번째는 Relative Error(RE) 이다.

일반적으로 ATE[17]가 대표적으로 성능 비교의 척도로 사용한다. ATE를 계산하기 위해 단일 상태에서의 groundtruth X_i 와 \hat{X}_i 사이에 존재하는 오차를 다음과 같이 매개변수화한다.

$$\Delta x_i = \{\Delta R_i, \Delta P_i, \Delta V_i\} \quad (1.1)$$

ΔR_i 은 회전 오차, ΔP_i 은 위치 오차, ΔV_i 은 속도 오차를 의미한다. 그리고 다음 식을 만족한다.

$$\begin{aligned} R_i &= \Delta R_i \hat{R}_i \\ P_i &= \Delta R_i \hat{P}_i + \Delta P_i \\ V_i &= \Delta R_i \hat{V}_i + \Delta V_i \\ \Delta R_i &= R_i (\hat{R}_i)^T \\ \Delta P_i &= P_i - \Delta R_i \hat{P}_i \\ \Delta V_i &= V_i - \Delta R_i \hat{V}_i \end{aligned} \quad (1.2)$$

마지막으로 N개의 시스템 상태에 대해서 Root Mean Square Error(RMSE)가 계산된다.

$$\begin{aligned}ATE_{rot} &= \left(\frac{1}{N} \sum_{i=0}^{N-1} \|\angle(\Delta R_i)\|^2 \right)^{\frac{1}{2}} \\ATE_{pos} &= \left(\frac{1}{N} \sum_{i=0}^{N-1} \|\Delta P_i\|^2 \right)^{\frac{1}{2}}\end{aligned}\tag{1.3}$$

여기에서 \angle 은 rotation matrix을 오일러 각 표현으로 바꾸는 것을 의미한다.

제 2 장

자율 주차(Autonomous Parking)

2.1 자율 주차란?

자율주차(Autonomous Parking)는 자동차를 사람의 개입없이 자동적으로 주차공간에 평행주차, 수직주차, 사선주차 등의 방법으로 주차하는 것을 의미한다. 자율주차가 가능해지면, 기존의 많은 조종과 주의가 필요한 제한된 주차공간에서의 주차보다 향상된 안정성과 편의성을 지닌 주차를 할 수 있게 된다. 자율주차는 Advanced Driver Assistance Systems(ADAS) 라고 불리는, 운전자에게 주행상의 편의성과 안전성을 제공하기 위한 운전자 보조 시스템의 4단계에 속하는 기술이다. 일반적으로 ADAS는 6개의 단계로 나뉘는데 0단계부터 3단계는 차선 감지, 차선 유지, 충돌 가능성이 있는 장애물 감지, 긴급 정지 등의 기능이 주였다면, 자율주차가 포함된 4단계부터는 자동차의 제어의 많은 부분이 자율화된다. 현재에는 3단계의 상용화가 진행중이며, 4단계 기술은 활발히 연구되고 있다.

2.2 자율 주차의 발전

세계에서 자율 주차 연구의 첫번째 시도는 1990년대 중반, INRIA의 Ligier사 자율 주행 전기차를 이용한 자율 평행 주차이다[18]. 이러한 자율 주차의 시스템은 일반적으로 초음파 센서에서 방출된 음파가 주변 물체에

반사되서 돌아오는 신호를 감지하여 주변 환경을 인지하고, 조향각과 주행 속도를 제어해서 가능한 주차 공간에 계획된 경로를 통해서 자동차를 주차하는 것을 목표로 한다. 이후에도 이러한 초음파 센서 기반의 자율 주차 시스템[19] 연구가 진행되었다. 이외에도 4개의 광각카메라를 이용한 Around View Monitor(AVM)와 자동차 엔코더 등의 멀티 센서 기반의 자율 주차 시스템 연구[4] 또한 진행되었다.

2.3 상용 자율 주차

2003년부터 Toyota에서는 Intelligent Parking Assist(IPA)라는 이름으로 자율 평행 주차 기능이 제공되는 Prius Hybrid 차량을 상용 판매하였다. 2006년부터는 Lexus에 자율 평행 주차뿐만 아니라 사선 주차 기능도 추가되었고, 2009년에는 Ford에서 Active Park Assist라고 불리우는 시스템을 개발하여 링컨 모델에 추가되어 판매되었다. 그외에도 BMW, Mercedes-Benz 등의 자동차 회사들 또한 활발한 관련 연구를 통하여 자사 자동차에 자율 주차 기능을 탑재하였다. 2015년에는 그전까지의 운전자가 개입해야 하는 부분적 자율 주차 기능과는 다른 완전 자율화된 주차 기능이 Bosch에 의해 제안되었다. Bosch가 제안한 완전 자율화된 주차 시스템은 2018년에 실제로 Mercedes-Benz 박물관 관람객들을 위해서 서비스로서 제공되었다. 하지만 주차장 자체가 자동차의 위치, 다른 자동차와의 통신등, 안전한 자율 주차를 위한 IT 기술이 적용되어 있어야 한다는 한계점이 있다.

제 3 장

자율 주차를 위한 AVM 기반 Visual SLAM

3.1 서론

자율주행자동차를 이용한 주차를 위해서는 자동차의 현재 위치 그리고 자동차 주변의 환경을 지도로 생성할 수 있는 기술이 필요하다. 이러한 기술은 동시적 위치추정 및 지도생성(SLAM)이라고 불린다. 일반적으로 자율주차에 사용할 수 있는 SLAM 기술은 매우 다양하다. 이는 SLAM 알고리즘에서 사용하는 센서의 종류에 따라서 분류를 해볼 수 있다.

첫번째는 흔히 어쿠스틱 센서라고 불리는 초음파 센서이다. 초음파 센서를 이용한 SLAM은 음파 신호를 주변으로 발산하여 주변의 물체에 반사되어 돌아오는 신호를 감지해서 자신의 위치와 주변 환경을 지도로 만든다. 이러한 방식은 주변의 물체가 있을 때는 잘 작동할 수 있지만, 주변의 공간이 비어있거나 주변에 물체가 있더라도 물체의 표면이 충분한 반사성을 가지고 있지 않거나 또다른 신호의 간섭에 의해 센서가 잘못된 신호를 감지할 수 있다는 불안정성이 있다[19]. 두번째로는 라이다 센서가 있다. 라이다는 Light Detection and Ranging(LiDAR)의 줄임말로, 광원과 수신기를 사용하여 원격의 개체를 탐지하고 거리를 측정하는 센싱 기술이다. 이 센서를 기반으로 하는 SLAM 알고리즘은 일반적으로 단일 센서를 이용하는 SLAM 알고리즘들중에서 가장 성능이 좋은 알고리즘에 속한다. 대

표적인 알고리즘으로는 Lidar Odometry and Mapping(LOAM)[9]이 있다. 하지만 이렇게 라이다를 기반으로 하는 SLAM 알고리즘의 성능이 좋고 하더라도 라이다 센서 자체의 비용이 다른 센서들보다 월등히 비싸고, 차량 자체에 기본적으로 부착되어 나오지 않기 때문에 일반적인 자율주차에 사용하기 힘든점이 있다. 그에 따라서 값싼 센서중에 하나인 카메라를 기반으로 하는 SLAM 기술의 연구가 활발히 이루어져왔다. 자율 주차 SLAM 기술에 사용할 수 있는 카메라 센서로는 일반적인 전방 카메라, 자동차에 앞, 뒤, 양옆에 각각 광각 카메라를 장착한후 자동차 주변 360도 환경을 하나의 이미지로 볼 수 있게 해주는 Around View Monitor(AVM) 센서 시스템이 있다. 전방 카메라 기반의 SLAM 기술은 자동차를 기준으로 자동차의 앞의 환경을 촬영하여 얻은 이미지를 이용한다. 이러한 전방 카메라 기반의 SLAM 기술에 대표적인 것으로는 모노 카메라를 사용하는 Direct Sparse Odometry(DSO)[2], 스테레오 카메라를 사용하는 ORB SLAM2[1], Stereo Direct Sparse Odometry(Stereo DSO)[3]가 있다. DSO와 같은 모노카메라를 사용하는 알고리즘같은 경우에는 카메라를 하나만 사용하기 때문에 SLAM을 통해 만든 지도의 스케일을 결정 할 수 없는 문제가 발생한다. 이는 ORB SLAM2 나 Stereo DSO와 같은 스테레오 카메라를 이용하여 해결되었는데, 카메라 두 대를 나란히 앞을 보도록 설치하고 시차가 존재하는 두 장의 이미지를 같은 시각에 촬영함으로써 SLAM을 통해 만드는 지도의 스케일을 결정할 수 있도록 하였다. 이러한 전방카메라 기반의 SLAM 기술은 카메라가 촬영하는 주변의 환경이 텍스처가 풍부하거나, 물체가 다양한 환경에서 가장 잘 작동을 한다.[20] 이는 전방카메라 기반의 SLAM은 이미지에서 고유의 특징을 가지는 ORB[10], BRIEF[12], SIFT[21] 와 같은 feature descriptor 또는 이미지의 픽셀 강도를 이용하여 자동차의 동작을 추정하기 때문이다. 실제로 자율주차를 하는 주차창

환경은 텍스처가 풍부하지 않고 대부분의 주차장이 야외에 있기 때문에 강렬한 빛이 카메라의 정상적인 작동을 방해할 때가 많다. 또한 주차의 이동경로에는 잦은 전후진과 빠른 회전이 포함되어 있기 때문에 시야각이 좁은 전방 카메라는 특징점을 놓쳐버릴 가능성이 높고[22], 결과적으로 앞서 언급한 주차장의 환경에서는 낮은 성능을 낼 가능성이 높다.

이에 비해서 AVIM 센서는 주변환경을 멀리 볼 수는 없지만 자동차 주변 360도를 모두 이미지로 얻을 수 있다. 또한 AVIM을 구성하는 카메라들은 일반적으로 땅바닥을 바라보도록 설치가 되어있기 때문에 자연광으로 인한 빛 번짐에 강건한 특징을 보인다. 하지만 이러한 AVIM의 전방 카메라에 대한 강점에도 불구하고 최근에 진행된 자율주차를 위한 AVIM 기반의 SLAM 연구들은 자동차의 위치 추정은 라이다[23], 휠 엔코더, IMU[5]를 이용하고, 지도 생성이나 루프 결함을 위해 AVIM 이미지를 Image segmentation [5]이나 포인트 클라우드화 [23] 하여 사용을 하였고 위치 추정 자체에는 주요하게 사용하지 않았다. 반면에 전방 카메라의 경우 카메라를 이용한 위치 추정과 IMU와 같은 관성 측정 장치의 결합을 통하여 진행된 연구[24], [25], [26]가 있는데, 이 연구들에서는 실제로 카메라를 이용하여 얻은 위치추정정보와 추가적인 센서를 이용하여 얻은 위치추정정보를 결합하였을 때 각각 따로 사용하였을 때보다 높은 성능을 내는 것을 확인할 수 있었다. 따라서 AVIM 이미지를 이용하여 자동차의 위치에 대한 추정을 할 수 있다면 [23], [5] 과 같이 AVIM 과 다른 센서를 결합하여 알고리즘을 구성할 때, 전방 카메라의 경우와 비슷하게 더 정확하고 안정적인 성능의 SLAM 시스템을 구성할 수 있을 것이다. 또한 센서 결합을 하지 않더라도 AVIM만을 이용하여 주차장에서 전방카메라 기반의 Visual SLAM보다 강건하고 정확하게 자동차의 위치추정을 할 수 있다는 것은 큰 장점이 될 것이다. 이에 따라서 본 연구에서는 순수하게 AVIM 이미지를 이용한

Visual SLAM 알고리즘을 제시할 것이다. 본 연구에서는 자동차의 동작 정보를 알아내기 위하여 AVM 이미지에서 주차선의 교차점을 탐색하고 추적한다 그리고 이미지 내부의 모서리 정보를 추출하였다. 주차선의 교차점은 Yolo v3[6]를 사용하여 탐색하였고, Deep SORT[7]를 이용하여 추적하였다. 이미지 내부의 모서리는 Canny Edge Detection[27]을 사용하여 추출하였다. 또한 추출된 모서리 정보는 마치 라이다 센서로 주변을 스캔한 포인트 클라우드로 생각할 수 있는데, 일반적으로 이러한 포인트 클라우드 집단이 여러개가 있을 때 집단끼리의 상대적 위치를 결정하는 것을 스캔 매칭이라고 한다. 전통적으로 스캔 매칭은 1992년에 Besl 과 McKay가 제안한 Iterative Closest Point(ICP)[28]을 이용하여 해결된다. 비록 ICP 알고리즘이 간단하고 복잡성이 낮아서 매우 인기있게 사용되어 왔지만, 알고리즘이 초기값에 따라서 local extremum에 빠진다거나 많은 반복횟수로 인하여 지연되는 문제점이 있다. 이외에도 Hough Transform 을 이용한 방법[29], 히스토그램을 이용하는 방법[30], [31]이 있다. AVM 은 Top View 이미지를 제공하기 때문에 해당 이미지에서 추출한 스캔 데이터는 자동차 주변의 환경을 위에서 바라본 하나의 이미지라고 생각할 수 있고 주차 위치 추정 문제는 2D 위치 추정문제로 생각할 수 있다. 따라서 본 연구에서의 스캔 매칭 문제는 이미지 등록(Image Registration) 문제로 치환될 수 있다[32]. 일반적으로 초기값을 몰라도 두 이미지를 등록할 수 있는 많은 방법들이 있지만 스캔 이미지는 조밀하지 않은 점들의 집합이기 때문에 일반적인 이미지와는 성질이 다르다. 따라서 ORB[10], BRIEF[12], SIFT[21]와 같은 특징점을 기반으로 하는 이미지 등록방법은 적합하지 않다. 이미지 내의 모든 정보를 이용하여 이미지 사이의 회전, 이동을 추정해야 하기 때문에 이미지를 주파수 영역으로 변환함으로써 푸리에 변환을 이용하여 이미지 사이의 회전, 이동을 추정해낼 수 있다.

여기서 얻은 동작 정보와 앞서 얻은 교차점 정보를 이용하여 간단한 Nearest Neighbor 알고리즘을 통해, 지도에 등록되어 있는 교차점과 현재 관찰하는 교차점의 관계를 추정해내는 Data Association 과정을 진행한다. 그리고 나서 최종적으로 앞서 계산한 모든 정보들(교차점 위치, 추적 정보, 동작 추정 정보)을 이용하여 자동차의 최종적인 위치 정보를 재추정하고 갱신한다. 제안하는 Visual SLAM 알고리즘을 소개하기 위해 전체 구성을 다음과 같이 하였다. 3.2에서 제안하는 Visual SLAM의 방법에 대해서 설명하고, 3.3에서 제안한 방법의 타당성을 검증하기 위한 실험을 진행하였다. 또한 제안된 방법의 위치 추정의 정확도는 기존의 전방카메라 기반의 Visual SLAM의 결과에 대해서 LOAM[9]을 groundtruth로 하여 비교하였다. 마지막으로 3.4에서 앞서 진행한 실험에 대해서 고찰을 하고 결론을 짓고 마무리한다.

3.2 방법

3.2.1 제안하는 SLAM 파이프라인

제안하는 SLAM 알고리즘의 전체 파이프라인은 그림 1과 같다. 알고리즘이 진행되는 전체적인 순서는 다음과 같다. 첫번째로, AVM으로부터 Top View 이미지를 받게되면 미리 학습된 Yolov3[6]를 이용하여 교차점을 탐색한다. 탐색된 교차점은 Deep Sort[7]를 이용하여 빠르게 추적되고 고유의 ID가 부여된다. 이 ID는 추후에 Data Association 단계에서 더욱 정확한 추적을 위해 사용된다. 그리고 푸리에 변환을 이용하여 위치를 추정하기 위해 이미지의 모서리를 추출하게 되는데 이 때에는 Canny Edge Detection[27] 과 추가적인 노이즈, 색채 필터링을 통해서 모서리를 추출한다. 이러한 과정을 통해 AVM 이미지는 바이너리 흑백 이미지로 바뀌게 된

다. 교차점과 모서리 추출이 완료되었으면, 푸리에 변환을 통해 두 시점의 바이너리 이미지 사이에 동작 정보를 추출하게 된다. 푸리에 변환을 통해 얻을 수 있는 것은 두 이미지 사이의 회전, 이동 동작 정보이다. 또한 푸리에 변환을 통해 동작 정보를 추출하는 과정에서 발생하는 추정오차, 즉 불확실성을 미리 정의된 모델에 의해서 정량적으로 측정한다. 이렇게 측정된 불확실성을 이용하여 특정 조건을 만족하면 현재 이미지를 Keyframe 으로서 추가를 하는 과정을 거친후 다음 단계로 진행한다. 이렇게 교차점의 탐색, 추적, 동작 추출과 Keyframe 생성까지 완료가 되었으면 그 다음 단계는 푸리에 변환을 통해 얻은 동작정보와 앞서 Yolov3[6] 와 Deep Sort[7]를 통해 얻은 교차점 정보들을 토대로 새롭게 Data Association하는 것이다. 이 과정을 통해서 똑같은 landmark에 할당된 ID의 개수는 줄이고 이제까지 보지 못한 landmark정보는 새롭게 추가하여 단계를 마친다. Data Association까지 모두 완료가 된 후에 마지막 단계는 교차점을 이용한 landmark 기반의 localization이다. 이 단계는 앞서 Data Association 과정을 통해 얻은 현재 관찰된 교차점과 지도상의 교차점의 관계를 이용하여 현재 자동차의 위치를 추정하는 단계이다. 이 방법은 과거에 지도에 등록되었던 교차점의 위치 정보를 이용하므로 푸리에 변환을 통해 얻은 동작 정보보다 불확실성이 낮은 동작 정보를 얻을 수 있다는 장점이 있다. 하지만 Data Association 단계에서 최소 2개이상의 연관된 교차점 쌍이 존재해야 이 단계가 진행이 되고, 만약 2개 미만의 교차점 쌍이 존재한다면 푸리에 변환을 통한 동작 정보를 이용하여 현재 자동차의 위치 정보를 업데이트하게 된다.

제안하는 알고리즘의 파이프라인

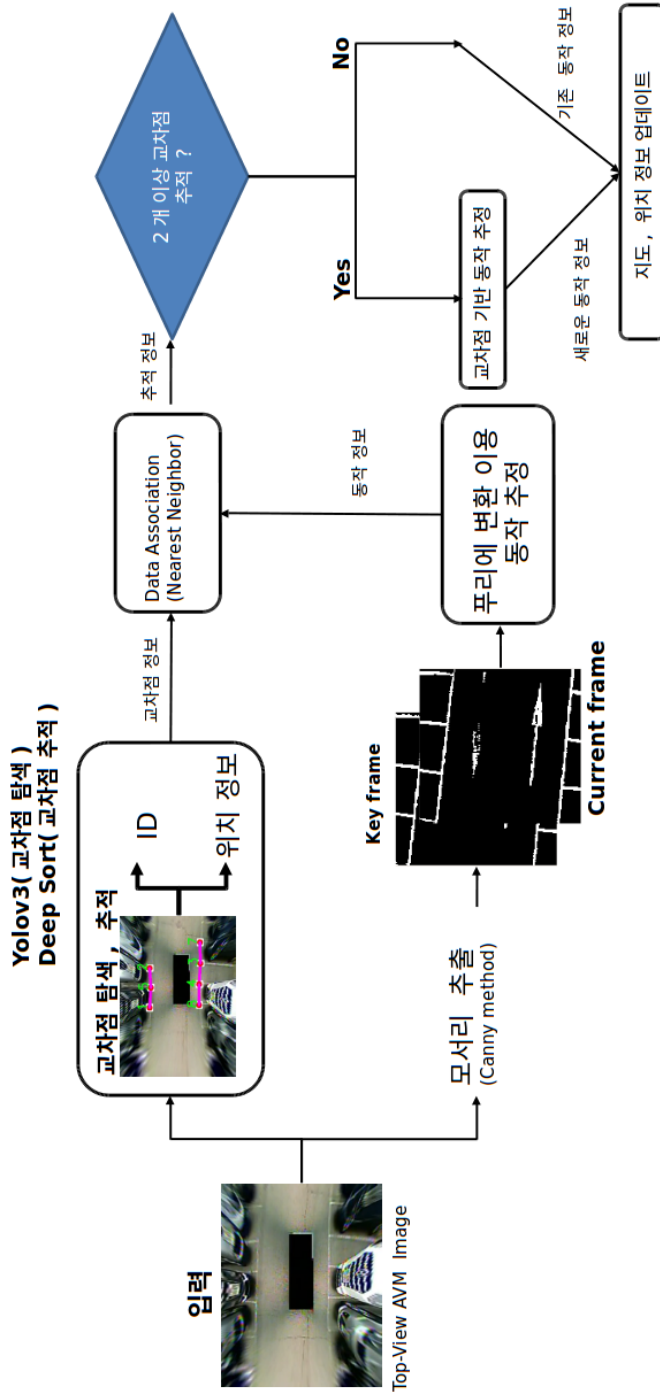
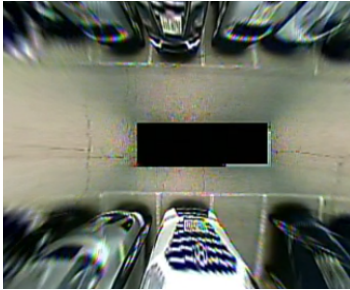


그림 1: 제안하는 SLAM 알고리즘의 전체 파이프라인

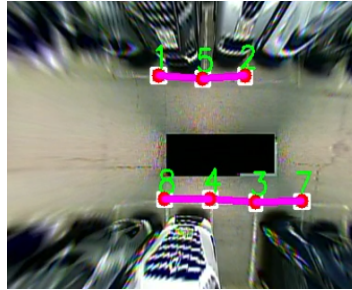
3.2.2 교차점의 탐색과 추적 그리고 모서리 추출

본 연구에서는 교차점을 탐색하기 위해서 CNN(Convolutional Neural Network)를 이용한 물체 탐색 방법인 Yolov3[6]을 사용하였다. 많은 물체 탐색 방법 [33], [6], [34]이 있지만, Yolo v3[6]가 일반적으로 속도는 초당 45프레임, 정확도는 다른 방법들보다 더 좋은 측면을 가지고 있기 때문에, Yolo v3[6]를 교차점 탐색을 위해 사용하였다. 또한 탐색된 교차점을 추적하기 위해서 Deep SORT[7]를 사용하였다. 물체 탐색 네트워크는 주차장에서 촬영된 AVIM 이미지를 가지고 훈련이 되었고, 입력으로 AVIM 이미지를 받으면 출력으로서 이미지상에 교차점이 존재할 수 있는 영역을 나타내는 사각형 형태의 네 점의 위치를 얻을 수 있다. 본 연구에서는 이 네점의 중앙점을 교차점의 위치로서 사용을 하였다. 물체 탐색 네트워크를 통해서 AVIM 이미지에서 교차점을 탐색하게 되면, 탐색된 교차점의 정보를 Deep SORT[7]에 입력으로 주어서 최종적으로 교차점의 위치와 고유 ID를 출력으로 얻게된다. 입력으로 사용되는 AVIM 이미지는 그림 2-(a) 이고, 출력된 교차점 정보들을 입력 이미지에 표시한것이 그림 2-(b)이다.

이와 동시에 입력 AVIM 이미지를 이용하여 이미지상의 모서리정보를 추출하였다. 이때 사용된 방법은 Canny Edge Detection[27]이고, 추가적으로 이미지의 노이즈를 제거하기 위해서 Gaussian Filtering을 진행하여 마무리하였다. 이때 출력으로 얻을 수 있는 이미지는 그림 3과 같이 추출된 모서리가 픽셀 강도 255의 흰색, 나머지 부분은 흑색으로 표현되어 있는 바이너리 흑백 이미지이고, 이 바이너리 흑백 이미지는 다음 3.2.3에서 설명할 푸리에 변환을 이용한 위치 추정에 사용된다.



(a) 입력 AVM 이미지



(b) 교차점을 AVM 이미지에 표시한 모습(흰색 사각형: 교차점이 존재할 수 있는 영역, 빨간색 점: 흰색 사각형을 이루는 네 점의 중앙점, 숫자: 교차점의 고유 ID)

그림 2: 교차점 탐색과 추적 결과

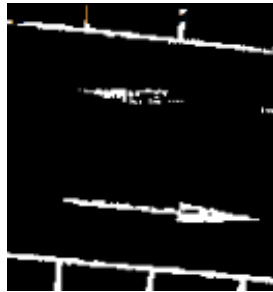


그림 3: AVM 이미지로부터의 모서리 추출 결과

3.2.3 푸리에 변환을 이용한 위치 추정

푸리에 변환은 이미지 등록을 위한 여러가지 성질들을 가지고 있다. 또한 주파수 영역을 사용하면, 저주파수 또는 고주파수 노이즈등의 제거를 통해 아주 강건한 특징을 보인다. 두 개의 이미지를 등록하는 간단하면서도 아름다운 방법은 위상 연관[35]이다. 위상 연관은 위상 전이 이론[36], [37], [38], [39]으로 더 잘 알려져 있는 푸리에 변환의 이동 성질을 이용한다. 이러한 성질은 다음과 같이 설명될 수 있다. 두 개의 함수 $f_1(x), f_2(x)$ 가 다음과 같은 관계를 가진다고 해보자.

$$f_2(x) = f_1(x - x_0) \quad (3.1)$$

여기서 x_0 는 두 함수사이의 평행이동값이다. 그리고 식 3.1 양변을 고속 푸리에 변환을 통해서 변환시킨다.

$$F_2(u) = F_1(u)e^{-j2\pi ux_0} \quad (3.2)$$

따라서 두 함수 f_1, f_2 사이의 위상 차이는 다음과 같은 cross power spectrum으로 정의된다.

$$e^{-j2\pi ux_0} = \frac{F_1(u)F_2(u)^*}{|F_1(u)F_2(u)|} \quad (3.3)$$

여기에서 *은 켈레 복소수 연산자이다.

식 3.3의 역 푸리에 변환은 두 이미지 사이의 평행이동값을 중심으로하는 델타 함수이다. 이 델타함수의 크기가 가장 큰 위치의 인덱스가 결과적으로 얻을 수 있는 평행 이동 값이다. 따라서 이미지 등록에 있어서 푸리에 변환의 이러한 성질들을 이용하기 위해서 이미지를 이에 맞게 변환하는 과정이 필요하다. 먼저 회전 이동 값을 추정하기 위한 방법을 설명한다.

푸리에 변환을 이용하여 회전 이동 값을 얻기 위해서는 회전량을 평행 이동으로 바꾸는 작업이 필요하다. 이러한 작업은 일반적으로 Cartesian 좌표계를 극좌표계(polar coordinate)로 바꿈으로써 가능하다. Cartesian 좌표계에서의 회전은 극좌표계에서의 평행이동과 같기 때문이다. 해당 내용을 수식으로 표현하면 다음과 같다. 초기에 두 이미지 $f_1(x,y), f_2(x,y)$ 가 있다고 해보자. 두 이미지는 서로 회전 α , 평행이동 x_0, y_0 의 관계에 놓여있다고 하면 다음과 같이 표현할 수 있다.

$$f_1(x,y) = f_2[(xcos\alpha + ysin\alpha) - x_0, (-xsin\alpha + ycos\alpha) - y_0] \quad (3.4)$$

이후 식 3.4의 양변의 푸리에 스펙트럼을 구한다. 이는 푸리에 스펙트럼이 이동 변환에 있어서 영향을 받지 않기 때문에, 각 이미지 사이의 존재하는 평행 이동의 영향을 없애기 위한 작업이다. 그렇게 되면, f_2 에 존재했던 평행이동 값 x_0, y_0 는 사라지고 다음의 식으로 정리가 된다.

$$|F_1(u,v)| = |F_2(ucos\alpha + vsin\alpha), (-usin\alpha + vcos\alpha)| \quad (3.5)$$

식 3.5과 같이 얻어지는 푸리에 스펙트럼을 극좌표계로서 나타내게 되면

$$F_1(\rho, \theta) = F_1(\rho, \theta - \alpha) \quad (3.6)$$

의 관계로 정리가 되게 된다. 식 3.6는 두 함수간에 평행이동만이 존재하는 관계이므로 식 3.1, 3.2, 3.3의 과정을 거쳐서 α 를 구할 수 있다. 회전을 구하는 것이 완료가 되면, 다음은 이미지 사이의 평행이동 값을 구하는 것이 남았다. 이미지의 평행이동의 측정은 한 스캔 이미지를 앞서 구한 회전 값 α 만큼 회전시켜서 두 스캔 이미지 사이의 회전 관계를 없앤 후 진행한다.

이 과정을 거친후, 가로와 세로 방향의 평행이동 값을 구하기 위해서 두 개의 스캔 이미지를 각각 수평 방향, 수직 방향으로 사영한다. 이를 수식으로 표현하면 다음과 같다. $f_2(x,y)$ 이미지를 회전 값 α 만큼 회전하여 얻은 이미지를 $f_3(x,y)$ 라고 하고, 이미지의 너비를 w , 높이를 h 라고 하면,

$$f_3(x)_x = \sum_{y=0}^{h-1} f_3(x,y) \quad (3.7)$$

$$f_3(y)_y = \sum_{x=0}^{w-1} f_3(x,y) \quad (3.8)$$

마찬가지로 이미지 $f_1(x,y)$ 에 대하여 각각 사영된 결과는 다음과 같다.

$$f_1(x)_x = \sum_{y=0}^{h-1} f_1(x,y) \quad (3.9)$$

$$f_1(y)_y = \sum_{x=0}^{w-1} f_1(x,y) \quad (3.10)$$

사영된 결과를 이용하여 식 3.1, 3.2, 3.3의 과정을 통하여 두 이미지 사이의 평행이동 관계를 추정할 수 있다.

3.2.4 Keyframe 생성

Keyframe이란 본디, 동영상 압축에서 비슷한 여러 이미지를 모두 저장하지 않고 하나의 대표하는 이미지로 저장하여 저장공간을 절약하기 위해서 사용되었던 개념이다. Visual SLAM에서는 대표적으로 Parallel Tracking And Mapping(PTAM)[40]에서 사용하였고, Keyframe을 사용함으로써 지도에 저장해야하는 특징점의 수를 현저하게 낮출 수 있었다. 결

과적으로는 계산비용, 저장공간의 절약이 가능하였다. 또한 중요한 것중에 하나는, Keyframe에 대해서 Tracking을 하게 되면 Keyframe과 현재 이미지 사이에 존재하는 다른 이미지들의 Tracking 과정에서 발생하는 위치 추정 오차를 무시할 수 있게 된다. Keyframe을 설정하는 기준은 Visual SLAM이 사용되는 환경, 알고리즘의 구성에 따라 다양하게 바뀔 수 있다. 일반적으로 규칙적으로 특정 frame 수마다 정할 수 있고[41], [40], 두 개의 이미지에 보이는 특징점 개수가 특정 개수 이상일때[42] 정하는 등의 방법이 존재한다. 본 논문에서는 Keyframe을 생성하기 위하여 3.2.3 푸리에 변환을 이용한 위치 추정과정에서 발생하는 불확실성을 모델링하여 Keyframe 생성 기준으로 삼았다. 3.2.3의 두 이미지의 위상 차이인 식 3.3의 역 푸리에 변환의 결과는 델타 함수이다. 이 델타 함수의 최대값(peak value)는 이론적으로 두 이미지가 완벽하게 서로 등록(Registration)된다면 1.0 이어야 한다. 하지만 두 이미지가 서로 다른 부분, 그리고 이미지에 존재하는 노이즈로 인해 최대값은 감소하게 된다[36]. 또한 델타 함수에 최대값과 가까운 값이 많을수록 이미지 등록의 확실함은 떨어지게 된다. 따라서 본 논문에서는 푸리에 변환을 이용한 이미지 등록의 이러한 특징을 이용하여 불확실성을 정의하였고 구체적인 식은 다음과 같다.

$$Uncertainty = \frac{\text{델타 함수에서 Peak value} * k \text{ 이상의 값 개수}}{\text{Peak value}} \quad (3.11)$$

식 3.11에서 k는 0에서 1사이의 값을 정하였고, k가 0에 가까워질 수록, 불확실성이 쉽게 커져서 Keyframe 생성이 자주 발생하고, 반대로 k가 1에 가까워질 수록, 불확실성이 잘 커지지 않기 때문에 Keyframe 생성이 잘 발생하지 않는다. 따라서 0 과 1사이에서 적절한 값의 선택이 필요하다.

3.2.5 Data Association

Data Association이란 현재 관찰된 특징점을 지도상의 등록되어 있는 특징점과 연관짓는 작업을 말한다. Data Association이 올바르게 진행된다면, 현재 관찰된 특징점이 지도상의 어떤 특징점과 같은 특징점인지 파악할 수 있고, 결과적으로 연관지어진 지도상의 특징점에 대해서 현재 자동차의 위치를 특정지을 수 있다. 이러한 방식으로 Data Association을 통해서 자동차의 위치를 추정하는 것은 ORB SLAM2[1]과 같은 특징점 기반의 SLAM 알고리즘에서는 가장 필수적인 과정이다. 앞서서 Yolov3[6]를 이용하여 교차점을 탐색하였고, Deep Sort[7]를 통하여 탐색된 교차점을 추적하였다. 이렇게 추적된 교차점은 교차점의 위치와 고유한 ID를 할당 받게 되는데, 할당받은 ID가 지도에 등록되어 있는 교차점의 ID와 같다면 Data Association이 성공한 것이다. 하지만 Deep SORT[7]는 물체의 선형 속도 모델을 가정하는 SORT[16]를 CNN 기법을 사용하여 같은 landmark에 여러개의 ID가 할당되는 ID switching 현상을 개선하였지만, 교차점 정보는 실제로 각 교차점마다 생김새가 매우 비슷하기 때문에 Deep SORT가 제대로 된 성능을 내기 어렵다. 따라서 이러한 문제를 해결하기 위해서 본 연구에서는 푸리에 변환을 통해 얻은 동작 정보와 앞서 추적된 교차점 정보를 모두 이용하여 새로운 Data Association을 진행하였다.

일반적으로 Data Association은 센서의 관측 결과의 가능도(likelihood)를 최대화하는 Data Association n_t 을 결정하는 작업이다. 이는 다음과 같은 식으로 표현할 수 있다[43].

$$\hat{n}_t = \underset{n_t}{\operatorname{argmax}} p(z_t | n_t, \hat{n}^{t-1}, s^t, z^{t-1}, u^t) \quad (3.12)$$

식 3.12에서 아랫 첨자 t 는 시간 t 에서의 단일 상태, 윗 첨자 t 는 시간

t 까지의 모든 상태를 의미한다. n_t 는 시간 t 에 Deep Sort[7]로부터 들어온 Data Association 결과이고, \hat{n}^{t-1} 은 시간 $t-1$ 까지의 위의 최대 가능성도 측정자를 통해 추정된 모든 Data Association 결과, z_t 는 시간 t 의 Yolo v3[6]로부터의 탐색된 교차점의 위치, z^{t-1} 은 시간 $t-1$ 까지의 모든 교차점 탐색 결과, u_t 는 시간 t 에서의 푸리에 변환을 통하여 얻은 자동차의 동작 정보를 의미한다. 그리고 마지막으로 s^t 는 시간 t 까지의 자동차의 모든 위치 정보를 의미한다. 본 논문에서는 Nearest neighbor 방법을 이용하여 최대가능도 측정자의 Negative log likelihood를 거리 함수로서 설정하였다. 구체적으로는 푸리에 변환을 통하여 얻은 자동차의 동작 정보를 이용해 얻은 자동차의 월드 좌표계에 대한 현재 위치 상태를 반시계가 양수인 회전 α , 오른쪽 방향이 양수인 가로 평행이동 x_0 , 위쪽 방향이 양수인 세로 평행이동 y_0 이라고 하고 현재 관측된 교차점의 위치가 이미지의 중심을 원점으로 하여 오른쪽 방향이 양수인 너비 방향의 위치를 u , 위쪽 방향이 양수인 높이 방향의 위치를 v 라고 할 때, 관측된 교차점의 위치를 자동차의 동작 정보를 이용하여 월드 좌표계로 좌표변환한다. 이 과정은 다음과 같이 표현된다.

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \quad (3.13)$$

Nearest neighbor방법(k -d tree[44])과 식 3.13을 통해 얻은 관측된 교차점의 월드 좌표계에서의 위치 x, y 를 이용하여 지도상에서 위치 x, y 와 가장 가까이 있는 교차점을 탐색한 후에 일정 이하의 거리(Euclidean distance)를 만족한다면, 계산된 정보를 새로운 Data Association 정보 (\hat{n}_t)으로 추가한다.

3.2.6 교차점을 이용한 Landmark 기반의 위치 추정

본 연구는 자동차의 위치를 결정하는 변수를 2차원의 회전, 가로 이동, 세로 이동, 총 3개로 가정하기 때문에 이 변수들을 결정하기 위해서는 최소 2개의 연관된(association) 교차점 쌍이 필요하다. 각 교차점 쌍마다 2개의 식을 얻을 수 있기 때문에 2개의 교차점 쌍을 알고 있다면 총 4개의 식을 세울 수 있고 최종적으로 위치와 관련된 변수를 결정할 수 있는 것이다. 따라서 앞선 Data Association 과정에서 얻은 교차점 쌍이 최소 2개 이상이라면 교차점을 이용한 Landmark 기반의 위치 추정이 이루어진다. 구체적인 위치 추정 방법은 다음과 같다. 교차점 쌍은 각 교차점을 나타내는 월드 좌표계에서의 교차점의 위치, 이미지 좌표계에서의 교차점의 위치로 이루어져 있다. 각 위치를 X_W, X_I 라고 한다. 새롭게 추정할 자동차의 회전, 이동을 나타내는 변수를 각각 R, t 라고 하고, R 은 이미지 좌표계의 점을 월드 좌표계로 나타내는 역할을 하고 t 는 월드 좌표계에서의 자동차의 가로, 세로 방향의 평행 이동을 나타낸다. 이를 구체적인 수식으로 나타내면 다음과 같다.

$$RX_I + t = X_W \quad (3.14)$$

다음으로는 N개의 교차점 쌍, X_I, X_W 의 centroid를 구한다.

$$\begin{aligned} centroid_{X_I} &= \frac{\sum_{i=1}^N X_I^i}{N} \\ centroid_{X_W} &= \frac{\sum_{i=1}^N X_W^i}{N} \end{aligned} \quad (3.15)$$

centroid를 구하는것이 완료되었으면 각 교차점 집합 X_I, X_W 에서 centroid를 빼준다. 이는 각 교차점 집합의 중심을 원점으로 보내주어서 평행 이동

변수 t 를 고려하지 않아도 되게 된다. 이후 Singular Value Decomposition을 통하여 회전 변수 R 을 구한다. 구체적인 수식으로 다음과 같이 표현된다.

$$H = (X_I - centroid_{X_I})(X_W - centroid_{X_W})^T$$

$$USV^T = H \quad (3.16)$$

$$R = VU^T$$

평행이동 변수 t 는 앞서 구한 회전 변수 R 을 이용하여 식 3.14에 대입하여 구하고 마무리한다. 최종적으로 모든 과정이 마무리 되면 현재 자동차의 위치와 지도를 갱신한다. 매번 이미지가 입력으로 들어올 때마다 이 과정들을 반복한다.

3.3 실험

3.3.1 실험 준비

앞서 제안한 AVIM 기반의 Visual SLAM 알고리즘의 타당성을 검증하기 위해서 실험을 계획하였다. 본 연구는 주차선이 보이는 주차장에서 자동차가 후방 직각 주차를 하는 상황을 가정을 하였다. 실험을 위해 서울대학교 동적로봇시스템 연구실의 자율주행용 자동차 시스템을 사용하였고, 해당 시스템은 그림 4와 같다. 구체적으로 사용한 센서는 다음과 같다. groundtruth 데이터를 만들기 위한 라이다 센서, 전방 카메라 Visual SLAM 알고리즘을 실험하기 위한 스테레오 카메라 센서, 제안된 AVIM 기반의 Visual SLAM 알고리즘을 실험하기 위한 AVIM 센서를 사용하였다. 그리고 이 센서 데이터들을 기록하기 위해서 Robot Operating System(ROS)[45]를 사용하였다. 또한 실험을 진행한 환경은 대한민국 경기도 수원시 소재

의 서울대학교 융합과학기술대학원에 있는 민원인 주차장을 이용하였다. 해당 주차장의 모습은 그림 5와 같다.

제안된 알고리즘의 타당성은 세가지 측면에서 검증을 해보았다. 제안된 알고리즘에서는 두 AVM 이미지와 푸리에 변환을 이용하여 얻을 수 있는 동작 정보의 불확실성을 정의하였다. 이렇게 정의된 불확실성 정보를 이용하여 Keyframe을 생성하였고 이를 통해 AVM 이미지의 매 입력마다 동작 정보의 불확실성이 누적되는 것을 방지하고자 하였다. 따라서 실제로 동작 정보의 정확도가 개선이 되는지 실제 실험을 통해 확인을 하였다. 그리고 두번째로, 제안된 알고리즘에서는 Deep SORT의 교차점 추적 결과에 추가적으로 Data Association 과정을 진행하였는데, 이는 하나의 교차점에 대해서 여러개의 ID가 중복해서 할당되는 것을 막기 위해 제안이 되었던 과정이다. 따라서 실제로 하나의 교차점에 할당되는 ID의 개수가 감소하였는지 실험적으로 확인하고 그로 인해 자동차의 위치 추정 정확도에 어떠한 영향이 있었는지 확인하였다. 마지막으로 기존의 Visual SLAM의 대표적인 알고리즘 ORB SLAM2[1], Stereo DSO[3]에 대해서 제안된 알고리즘의 성능을 비교해보았다. 실험 결과에서는 후방 직각 주차 두번의 시도에 대한 결과 그래프와 결과 통계치를 보인다. 그리고 통계치는 Absolute Trajectory Error[17]을 사용하여 계산하였다.

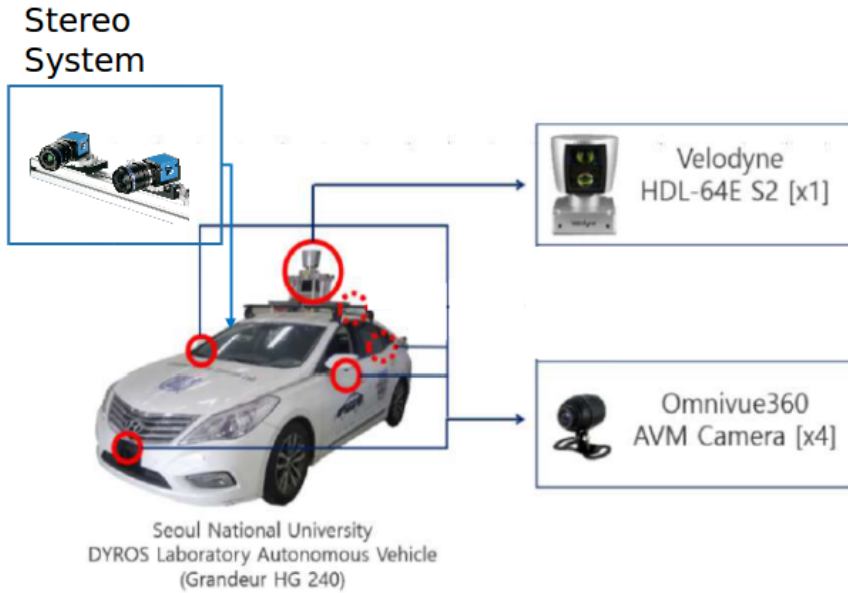


그림 4: 실험에 사용한 자율주행용 자동차 시스템



그림 5: 실험을 진행한 주차장, 융합과학기술대학원 민원인 주차장

3.3.2 실험 결과

- : 시작점 ● : 종료점
- : Ground truth(LOAM)
- : SLAM Path

그림 6: 실험 결과의 그래프에 쓰이는 표시의 의미

3.3.2.1 불확실성을 이용한 keyframe 생성의 효과

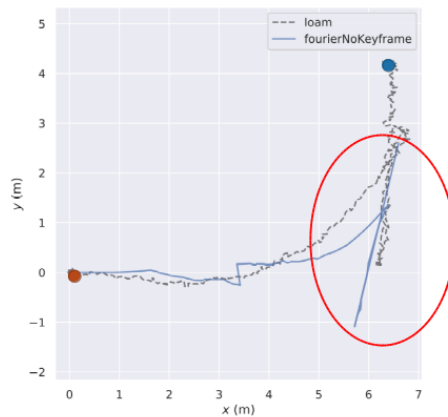


그림 7: Keyframe를 사용하지 않고 경로를 추정하였을 때의 결과. 각 그래프에서의 빨간색 타원은 전체 경로에서 회전 오차가 눈에 띄게 개선이 된 부분을 표시한 것이다.

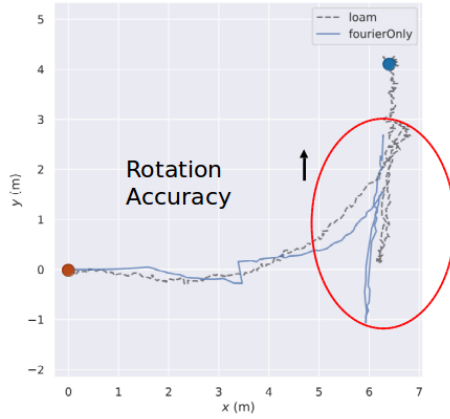


그림 8: Keyframe을 사용하여 경로를 추정하였을 때의 결과

3.3.2.2 동작정보를 이용한 Data Association의 효과

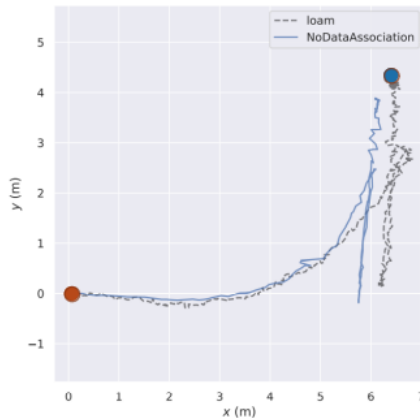


그림 9: 동작정보를 이용하여 Data Association을 하지 않을 때의 경로

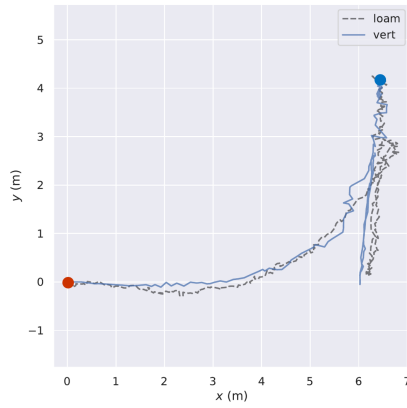


그림 10: 동작정보를 이용하여 Data Association을 하였을 때의 경로

	Deep SORT	Deep Sort + 추가 Data Association
교차점당 할당 된 ID의 평균 개수	9.1	1.5

표 1: 교차점당 평균 몇 개의 아이디가 할당되는지 계산한 결과

3.3.2.3 위치추정 정확도의 비교

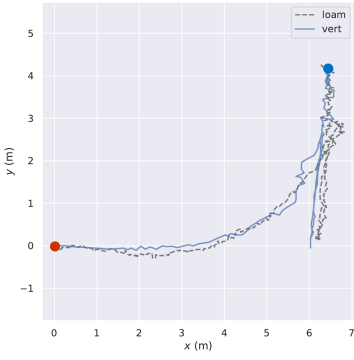
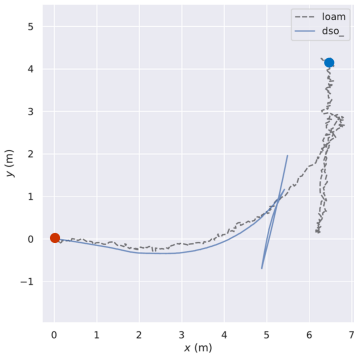
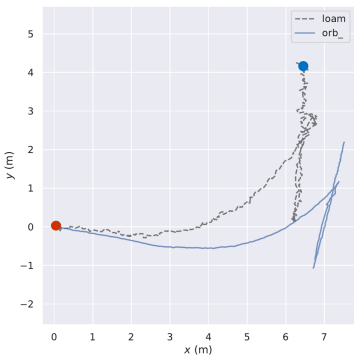
Method	min(m)	rmse(m)	max(m)	std(m)	path
Proposed	0.1480	0.5905	1.9415	0.3757	
Stereo DSO[3]	0.0001	1.9581	2.6999	0.8183	
ORB SLAM2[1]	0.0005	1.6596	2.4396	1.0124	

표 2: 주차 1 결과(Total length: 29.122m)

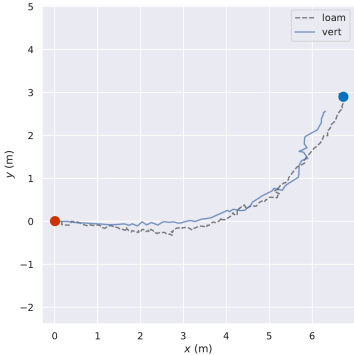
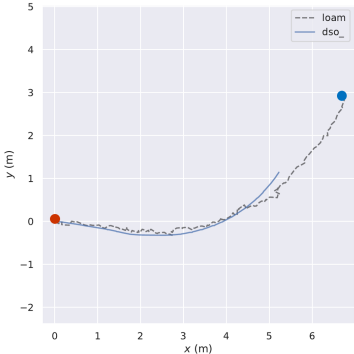
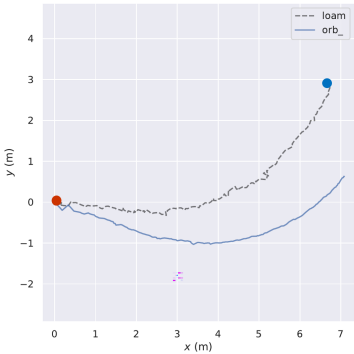
Method	min(m)	rmse(m)	max(m)	std(m)	path
Proposed	0.1406	0.4702	0.7888	0.2065	
Stereo DSO[3]	0.3484	1.3656	2.0061	0.4709	
ORB SLAM2[1]	0.6296	2.0696	2.9020	0.7075	

표 3: 주차 2 결과(Total length: 11.727m)

3.4 고찰 및 결론

3.4.1 고찰

3.3.2.1의 불확실성을 이용한 Keyframe 생성의 효과를 확인하는 실험으로부터, Keyframe을 생성하지 않고 동작 정보를 구했을 때 보다 Keyframe을 생성하여 동작 정보를 구했을 때 동작 정보의 정확성이 더 높은 것을 확인 할 수 있었다. 3.3.2.1의 빨간 타원 표시가 된 곳을 보면 자동차의 회전에 대한 추정이 Keyframe을 생성하지 않았을 때보다 groundtruth 데이터에 대해서 더욱 정확해진 것을 그래프 상에서 확인 할 수 있었다. LOAM[9]으로부터 회전 정보를 얻을 수 없었기 때문에 정확한 수치적 비교는 하지 못하였다. 하지만 그림 7의 경우 확연하게 경로의 방향이 틀어진 것을 확인 할 수 있고, 그림 8의 경우 그림 7에 비해 좀더 groundtruth 경로와 방향이 일치하는 것을 확인 할 수 있다.

두번째로는 동작 정보를 이용한 Data Association의 효과를 확인하는 실험을 하였다. 이 실험은 Deep SORT[7]의 결과에 푸리에 변환으로부터의 동작 정보를 이용하여 추가적으로 Data Association을 하였을 때 Deep SORT[7]만 사용하였을 때보다 ID switching 현상이 덜 발생하였다는 것과 제안된 알고리즘의 위치 추정 성능 향상되는 것을 확인하기 위해서 진행되었다. 실제로 표 1을 보면 Deep SORT[7]의 경우 교차점당 할당된 ID의 개수가 9.1 개였던 반면에 추가적인 Data Association을 진행하였을 경우 1.5개로 줄어드는 결과를 확인 할 수 있었다. 또한 이러한 ID switching 현상의 감소로 인하여 지도에 중복되는 교차점들의 정보가 쌓이는 것을 방지할 수 있고, 이로부터 추후에 잘못된 Data Association이 발생할 위험성을 줄이게 되었다. 결과적으로는 자동차의 동작 정보를 교차점을 이용하여 계산할 때 Data Association이 더 정확하게 되므로 위치 추정의

성능도 향상 되었다. 이러한 성능의 향상은 이미지에서 추출하는 교차점의 위치가 서로 멀다는 것과 푸리에 변환을 통해 특징점의 매칭없어도 자동차의 상대적인 움직임을 추정할 수 있기 때문인것으로 생각된다. 다만 이는 앞서 얻은 동작 정보가 오차가 클 때, 잘못된 Data Association이 발생할 수 있다. 이러한 문제는 AVM으로부터의 이미지에 왜곡이나 노이즈가 많을 때 발생할 수 있으며, 휠 엔코더와 같은 안정적인 센서를 기반으로 해결될 수 있을 것이다.

마지막으로 제안된 알고리즘과 전방 카메라 기반의 Visual SLAM의 성능을 비교해보았다. 후방 직각 주차 형태의 두번의 실험을 하였다. 각 실험은 주차 1, 주차 2으로 표현하였다. 두 번의 실험에서 rmse를 기준으로 비교군이 되는 기존의 전방카메라 기반의 Visual SLAM[1], [3] 대비 정확도가 향상된 것을 확인 하였다. 실제로 실험을 하는 환경은 빛이 내리쬐고 자동차가 많이 없고 주차선이 잘보이는 환경이었고, 주차 경로상에, 짧은 시간 동안 큰 회전이 포함된 주차 시나리오였는데, 전방 카메라 기반의 Visual SLAM 알고리즘의 경우에는 이러한 환경의 영향으로 성능이 하락하였다고 생각할 수 있다. 또한 AVM 센서 자체가 넓은 시야각과 빛 변화에 대한 강건함을 가지고 있고, 제안된 알고리즘에서는 주차선 상의 교차점과 주변 사물의 모서리 정보를 이용하여 위치추정을 하였기 때문에 주차장에서 좀 더 나은 결과를 얻을 수 있었던것으로 생각된다.

3.4.2 결론

본 연구에서는 AVM을 기반으로 하여 Visual SLAM 알고리즘을 제안하는 것을 목표로 삼았다. 결과적으로 본 연구를 전방 카메라 기반의 Visual SLAM과 비교를 했을 때 후방 직각 주차의 주차장 시나리오에서 더 나은 성능을 보이는 것을 실험적으로 확인했고, 제안된 각각의 방법들이

본 연구에서 위치 추정의 성능 향상에 의미가 있음을 실험을 통해 확인하였다. 비록 제안된 방법이 주차장에서 기존의 다른 Visual SLAM[1], [3]보다 나은 성능을 보였다고는 하지만, 실제로 자율주차문제는 센치미터 정확도의 위치 추정이 가능할 때 좀더 안정적인 주차가 가능하기 때문에 추후의 연구에서는 AVM과 초음파, IMU, 휠 엔코더, 라이다와 같은 물리적 센서들과의 융합이 시도되어야 할 것이다.

참고 문헌

- [1] R. Mur-Artal and J. D. Tardós, “Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras,” *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.
- [2] J. Engel, V. Koltun, and D. Cremers, “Direct sparse odometry,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 3, pp. 611–625, 2017.
- [3] R. Wang, M. Schworer, and D. Cremers, “Stereo dso: Large-scale direct sparse visual odometry with stereo cameras,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3903–3911.
- [4] C. Jang, C. Kim, S. Lee, S. Kim, S. Lee, and M. Sunwoo, “Re-plannable automated parking system with a standalone around view monitor for narrow parking lots,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 2, pp. 777–790, 2019.
- [5] T. Qin, T. Chen, Y. Chen, and Q. Su, “Avp-slam: Semantic visual mapping and localization for autonomous vehicles in the parking lot,” 2020.
- [6] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” 2018.
- [7] N. Wojke, A. Bewley, and D. Paulus, “Simple online and realtime tracking with a deep association metric,” in *2017 IEEE international conference on image processing (ICIP)*. IEEE, 2017, pp. 3645–3649.
- [8] R. O. Duda, P. E. Hart *et al.*, *Pattern classification and scene analysis*. Wiley New York, 1973, vol. 3.

- [9] J. Zhang and S. Singh, “Loam: Lidar odometry and mapping in real-time.” in *Robotics: Science and Systems*, vol. 2, no. 9, 2014.
- [10] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “Orb: An efficient alternative to sift or surf,” in *2011 International conference on computer vision*. Ieee, 2011, pp. 2564–2571.
- [11] E. Rosten and T. Drummond, “Machine learning for high-speed corner detection,” in *European conference on computer vision*. Springer, 2006, pp. 430–443.
- [12] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, “Brief: Binary robust independent elementary features,” in *European conference on computer vision*. Springer, 2010, pp. 778–792.
- [13] H. Zhou, D. Zou, L. Pei, R. Ying, P. Liu, and W. Yu, “Structslam: Visual slam with building structure lines,” *IEEE Transactions on Vehicular Technology*, vol. 64, no. 4, pp. 1364–1375, 2015.
- [14] R. L. Streit and T. E. Luginbuhl, “Maximum likelihood method for probabilistic multihypothesis tracking,” in *Signal and Data Processing of Small Targets 1994*, vol. 2235. International Society for Optics and Photonics, 1994, pp. 394–405.
- [15] J. Neira and J. D. Tardos, “Data association in stochastic mapping using the joint compatibility test,” *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6, pp. 890–897, 2001.
- [16] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, “Simple online and realtime tracking,” in *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016, pp. 3464–3468.
- [17] Z. Zhang and D. Scaramuzza, “A tutorial on quantitative trajectory evaluation for visual (-inertial) odometry,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 7244–7251.

- [18] I. E. Paromtchik and C. Laugier, “Motion generation and control for parking an autonomous vehicle,” in *Proceedings of IEEE international conference on robotics and automation*, vol. 4. IEEE, 1996, pp. 3117–3122.
- [19] M. Chirca, R. Chapuis, and R. Lenain, “Autonomous valet parking system architecture,” in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, 2015, pp. 2619–2624.
- [20] N. Yang, R. Wang, X. Gao, and D. Cremers, “Challenges in monocular visual odometry: Photometric calibration, motion bias, and rolling shutter effect,” *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 2878–2885, 2018.
- [21] D. G. Lowe, “Object recognition from local scale-invariant features,” in *Proceedings of the seventh IEEE international conference on computer vision*, vol. 2. Ieee, 1999, pp. 1150–1157.
- [22] M. Quan, S. Piao, M. Tan, and S.-S. Huang, “Accurate monocular visual-inertial slam using a map-assisted ekf approach,” *IEEE Access*, vol. 7, pp. 34 289–34 300, 2019.
- [23] G. Im, M. Kim, and J. Park, “Parking line based slam approach using avm/lidar sensor fusion for rapid and accurate loop closing and parking space detection,” *Sensors*, vol. 19, no. 21, p. 4811, 2019.
- [24] T. Qin, P. Li, and S. Shen, “Vins-mono: A robust and versatile monocular visual-inertial state estimator,” *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.
- [25] R. Mur-Artal and J. D. Tardós, “Visual-inertial monocular slam with map reuse,” *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 796–803, 2017.
- [26] A. Concha, G. Loianno, V. Kumar, and J. Civera, “Visual-inertial direct slam,” in *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2016, pp. 1331–1338.

- [27] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.
- [28] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Sensor fusion IV: control paradigms and data structures*, vol. 1611. International Society for Optics and Photonics, 1992, pp. 586–606.
- [29] A. Censi, L. Iocchi, and G. Grisetti, "Scan matching in the hough domain," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. IEEE, 2005, pp. 2739–2744.
- [30] T. Rofer, "Using histogram correlation to create consistent laser scan maps," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1. IEEE, 2002, pp. 625–630.
- [31] M. Bosse and J. Roberts, "Histogram matching and global initialization for laser-only slam in large unstructured environments," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, 2007, pp. 4820–4826.
- [32] G. Jiang, L. Yin, G. Liu, W. Xi, and Y. Ou, "Fft-based scan-matching for slam applications with low-cost laser range finders," *Applied Sciences*, vol. 9, no. 1, p. 41, Dec 2018. [Online]. Available: <http://dx.doi.org/10.3390/app9010041>
- [33] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *2008 IEEE conference on computer vision and pattern recognition*. IEEE, 2008, pp. 1–8.
- [34] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.

- [35] Y. Keller, A. Averbuch, and M. Israeli, “Pseudopolar-based estimation of large translations, rotations, and scalings in images,” *IEEE Transactions on Image Processing*, vol. 14, no. 1, pp. 12–22, 2004.
- [36] B. S. Reddy and B. N. Chatterji, “An fft-based technique for translation, rotation, and scale-invariant image registration,” *IEEE transactions on image processing*, vol. 5, no. 8, pp. 1266–1271, 1996.
- [37] J. N. Sarvaiya, S. Patnaik, and S. Bombaywala, “Image registration using log-polar transform and phase correlation,” in *TENCON 2009-2009 IEEE Region 10 Conference*. IEEE, 2009, pp. 1–5.
- [38] E. De Castro and C. Morandi, “Registration of translated and rotated images using finite fourier transforms,” *IEEE Transactions on pattern analysis and machine intelligence*, no. 5, pp. 700–703, 1987.
- [39] A. Behan, “2-d and 3-d image registration for medical, remote sensing, and industrial applications.” *The Photogrammetric Record*, vol. 21, no. 114, pp. 180–181, 2006.
- [40] G. Klein and D. Murray, “Parallel tracking and mapping for small ar workspaces,” in *2007 6th IEEE and ACM international symposium on mixed and augmented reality*. IEEE, 2007, pp. 225–234.
- [41] J. Stückler and S. Behnke, “Integrating depth and color cues for dense multi-resolution scene mapping using rgb-d cameras,” in *2012 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*. IEEE, 2012, pp. 162–167.
- [42] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, “Rgb-d mapping: Using depth cameras for dense 3d modeling of indoor environments,” in *Experimental robotics*. Springer, 2014, pp. 477–491.
- [43] M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit *et al.*, “Fastslam: A factored solution to the simultaneous localization and mapping problem,” *Aaai/iaai*, vol. 593598, 2002.

- [44] J. L. Bentley, “Multidimensional binary search trees used for associative searching,” *Communications of the ACM*, vol. 18, no. 9, pp. 509–517, 1975.
- [45] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, “Ros: an open-source robot operating system,” in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.

Abstract

Around View Monitor(AVM) Based Visual SLAM For Autonomous Parking

Chanwoo Ahn

Department of Transdisciplinary Studies

The Graduate School of Convergence Science and

Technology

Seoul National University

Autonomous parking consists of perception, planning, control. During perception procedure in autonomous parking, vehicle should know its location and perceive surrounding environment. This is called Simultaneous Localization And Mapping (SLAM). Many SLAM algorithms have been proposed for accurate perception of environment. Especially, Visual SLAM, which uses a cheap camera as a main sensor of SLAM algorithm, has been considered as promising algorithm for autonomous vehicle. Most of Visual SLAM use front camera setting. In this camera setting, Visual SLAM works well for most of environments. However, performance of the algorithm gets worse when environment has few features or strong sunlight condition. Most of parking lots are located outdoor and have monotonous features like parking

lines, cars. To address these problems and improve accuracy of Visual SLAM for autonomous parking, this paper proposes new Visual SLAM algorithm, which uses Around View Monitor(AVM) as a main sensor. As top-view images are generated in AVM system, fourier transform is used to extract motion information from the AVM images. Compared to traditional visual motion tracking methods[1], [2], [3] which use reprojection error or photometric error as a cost function to estimate motion, fourier transform can simply estimate motion from reference AVM image to target AVM image without any optimization or feature matching. Also, landmark based localization is used to estimate vehicle's motion more robustly and accurately. In this paper, landmark means cross points on parking lines. Landmark based localization in this paper consists of three procedure. First one is cross point detection. Cross points are detected using YoloV3[6]. Compared to other AVM based SLAM methods[4], [5] which use Image segmentation to detect features in parking lot, training procedure of the neural network is simpler and easier. Second one is data association. Data association means associating procedure among features in map and currently observed features in SLAM literature. Deep SORT[7] is used to track features using currently observed cross points. As re-identification of tracked features frequently occurs when using Deep SORT[7], additional data association is done using current motion estimation from image registration and currently observed cross points in Nearest Neighbor literature. As motion estimation accuracy from reference image to target image is considerably accurate and distance between cross points is far, data association accuracy is improved compared to the data association without this additional association procedure. After this data association, if the

number of associated features is larger than one, motion is newly estimated using Singular Value Decomposition and positions of associated features. To demonstrate improvement of proposed SLAM algorithm compared to other Visual SLAM algorithms, experiments in parking lot are suggested and compared with traditional Visual SLAM algorithms. Also, Lidar Odometry And Mapping(LOAM)[9] is used as a groundtruth for comparing the Visual SLAM algorithms.

Keywords : SLAM, Around View Monitor(AVM), Fourier Transform, Data Association, Landmark based localization, LOAM, Autonomous Parking

Student Number : 2019-21330