

자율주행 및 C-ITS

- SLAM 개요 -

수업 개요

2

□ 수업 내용

- ▣ 동시적 위치 추정 및 지도 작성(Simultaneous localization and mapping, SLAM) 개요
- ▣ 기존 SLAM
- ▣ SLAM의 어려움
- ▣ SLAM의 기술적 요소
- ▣ SLAM의 이해

□ 수업 목적

- ▣ 자율주행에 사용되는 SLAM의 개념에 대하여 이해

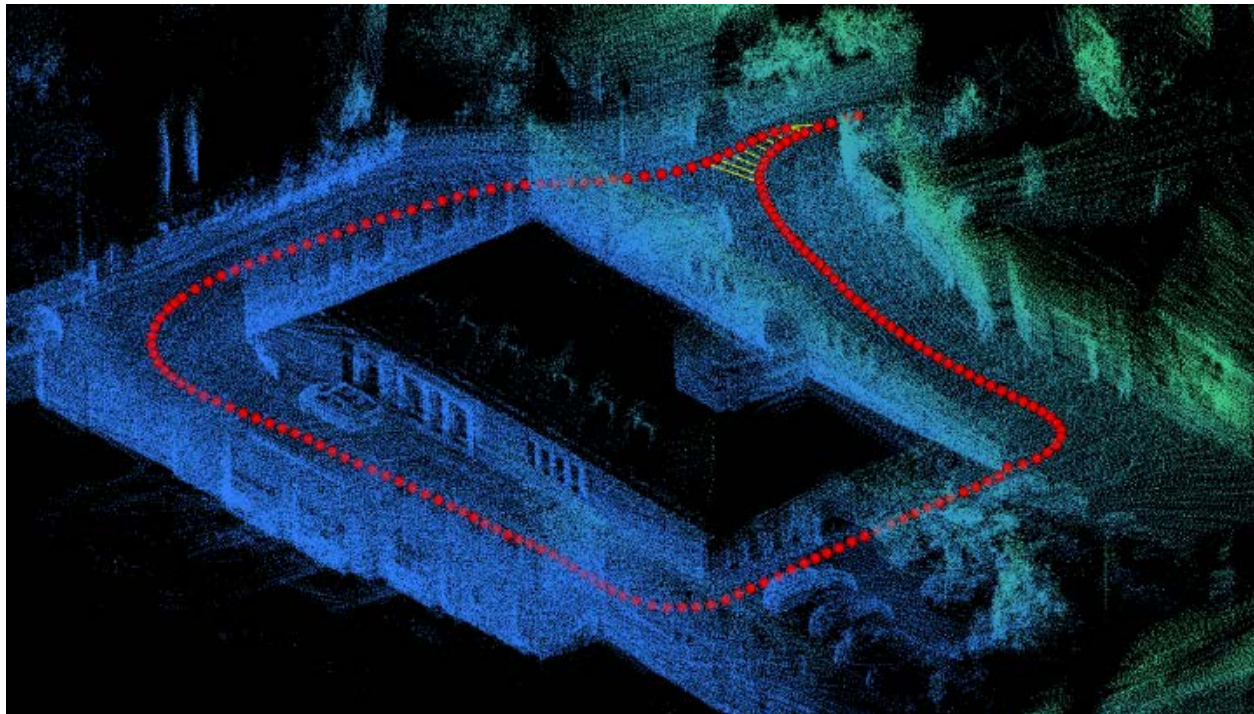
3

SLAM

SLAM 개요

4

- 동시적 위치 추정 및 지도 작성(Simultaneous localization and mapping, SLAM)
 - ▣ 자율주행 차량에 사용되어 주변 환경 지도를 작성하는 동시에 차량의 위치를 작성된 지도 안에서 추정하는 방법
 - ▣ 처음 보는 환경에서 위치 추정을 하기 위한 로봇틱스 기술



<출처: https://www.ifp.uni-stuttgart.de/en/research/photogrammetric_computer_vision/SLAM/>

SLAM 개요

5

□ SLAM의 정의에 대한 두 가지 견해 존재

▣ Cyrill Stachniss

- 연속된 실시간 센서 데이터로 위치 추정과 지도 작성을 동시에 하면 SLAM임
- Classical robotics 관점
- 많은 LiDAR SLAM / LiDAR-Odometry가 이 정의를 따름

▣ Davide Scaramuzza

- 연속된 실시간 센서 데이터로 위치 추정과 지도 작성을 동시에 하면서, 내가 만든 지도 안에서 항상 정확한 위치 추정이 되어야함 (i.e., Global consistency)
- Modern robotics 관점
- Large-scale mapping
- 최신 Visual-SLAM(카메라 기반)이 이 정의를 따름

SLAM 개요

6

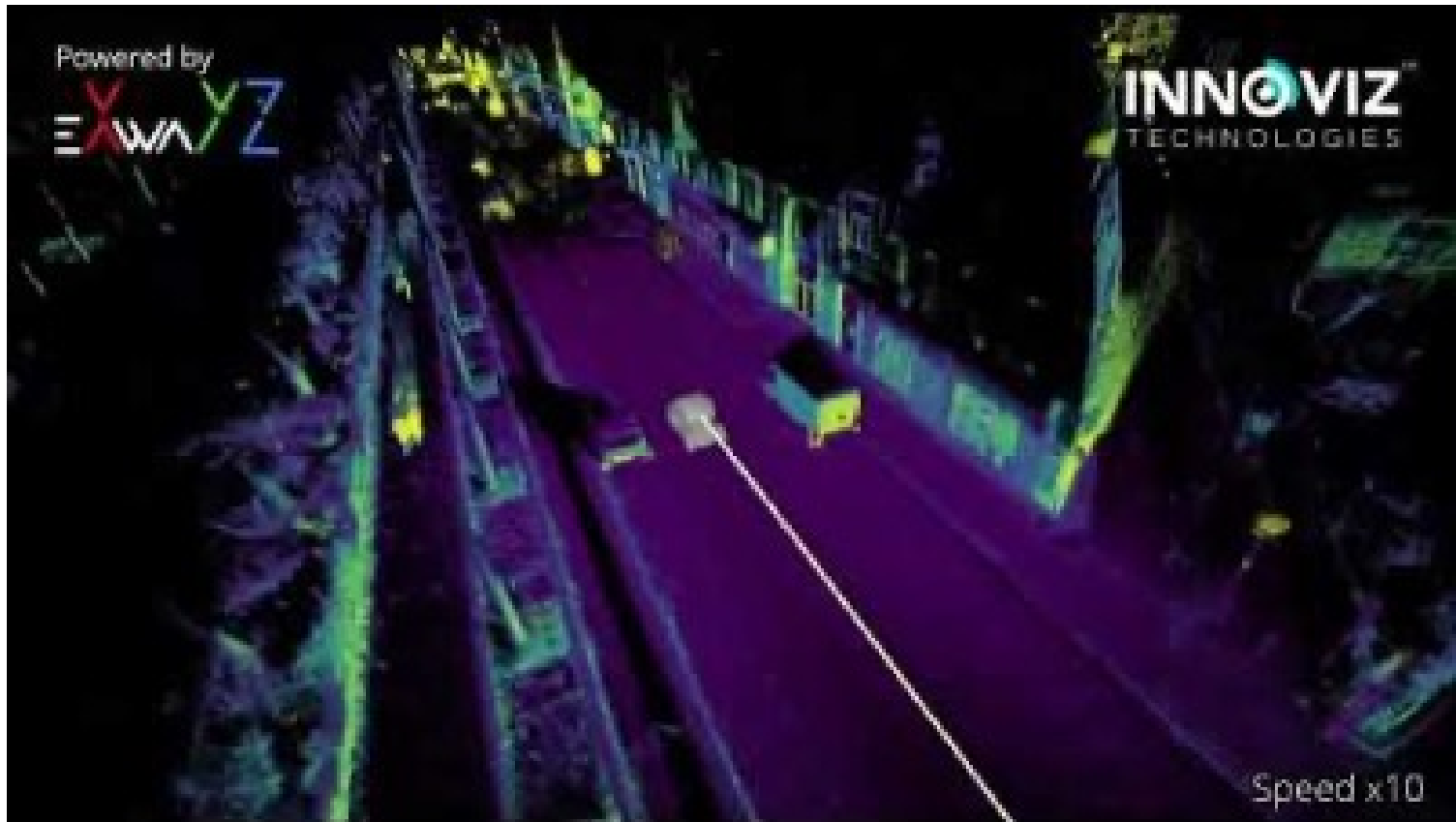
- SLAM 사용 사례
 - ▣ 실내 모바일 로봇



SLAM 개요

7

- SLAM 사용 사례 (계속)
 - ▣ 도심자율주행



SLAM 개요

8

- SLAM 사용 사례 (계속)
 - ▣ 도심자율비행

Robust Stereo Visual Inertial Odometry for Fast Autonomous Flight

**Ke Sun, Kartik Mohta, Bernd Pfrommer, Michael
Watterson, Sikang Liu, Yash, Mulgaonkar, Camillo J.
Taylor, Vijay Kumar**



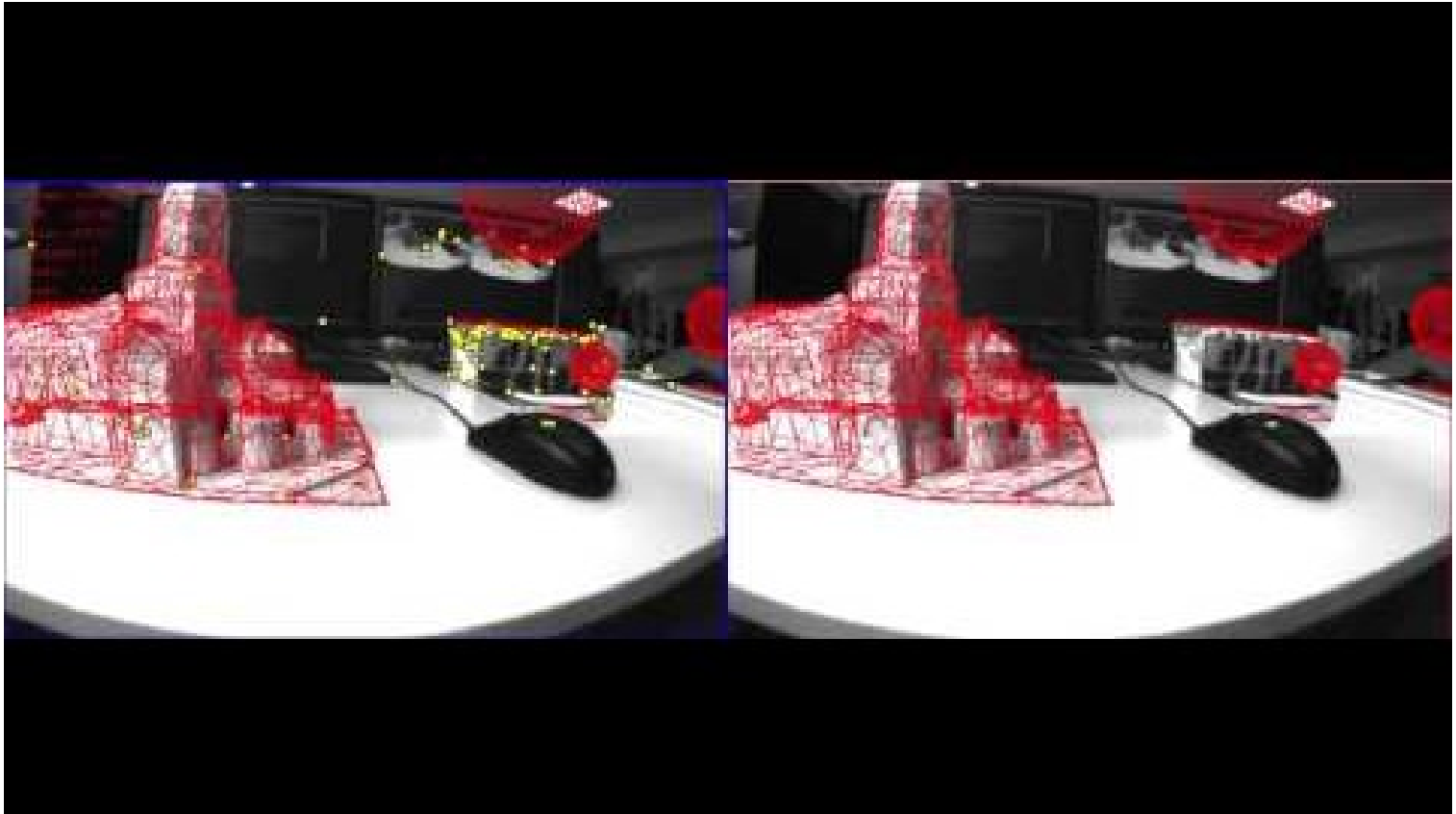
Penn | **GRASP**
Engineering | **Laboratory**

General Robotics, Automation, Sensing & Perception Lab

SLAM 개요

9

- SLAM 사용 사례 (계속)
 - ▣ VR / AR



기존 SLAM

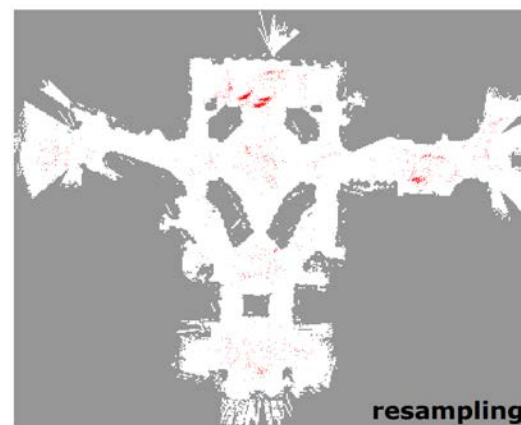
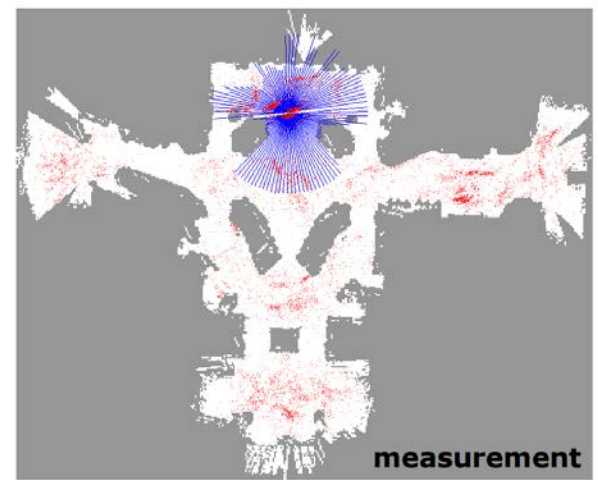
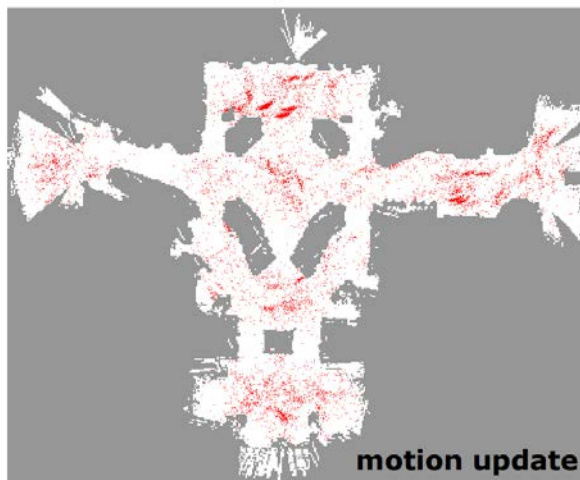
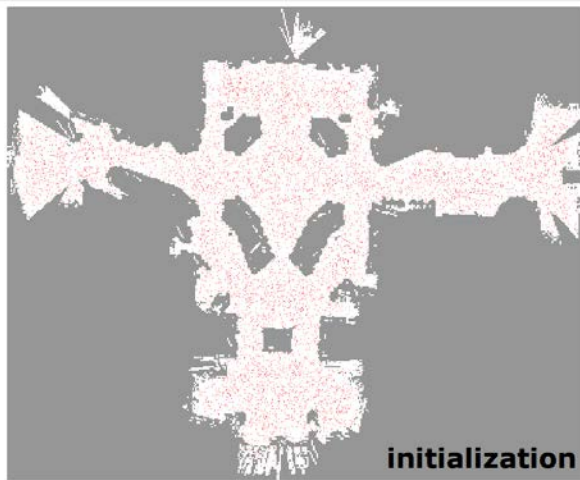
10

- 기존의 Localization 방식의 한계점
 - ▣ Monte-Carlo Localization (Particle Filter)
 - Configuration space (i.e., Map) 정보를 알 때, particle filter를 이용해서 위치 추정을 수행하는 방법
 - Map 의 정보가 정확하지 않을 때, 정확한 localization에 실패하게 됨
 - Map 이 없다면 사용할 수 없음
 - Discrete multi-modal 추정이기 때문에, 비슷한 맵 구간이 많으면 헛갈릴 수 있음

기존 SLAM

11

- SLAM 과정
 - ▣ 5개의 stage가 계속 반복됨



기존 SLAM

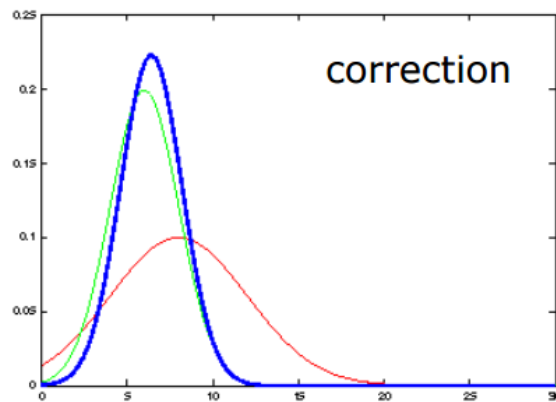
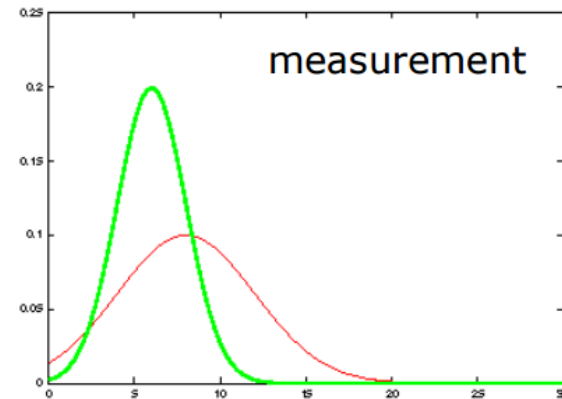
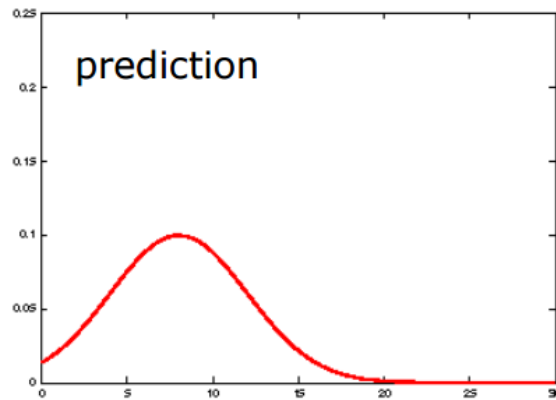
12

- 기존의 Localization 방식의 한계점 (계속)
 - ▣ Kalman filter (+ Extended Kalman filter)
 - Gaussian 분포의 motion model과 sensor measurement 값의 correlation을 구해 최적의 위치를 추정하는 방법
 - Kalman filter는 선형 시스템에서만 적용되기 때문에, 비선형 시스템을 선형 근사하는 Extended Kalman filter를 사용
 - 계산해야하는 state가 많아질수록 covariance matrix가 quadratic 하게 증가하기 때문에, 실시간 환경에서 모든 센서 데이터를 다 사용할 수 없음
 - EKF의 경우 선형근사를 하기 때문에, 매 state update 마다 에러가 쌓일 수 밖에 없음 (drift error)

기존 SLAM

13

- 기존의 Localization 방식의 한계점 (계속)
 - ▣ Kalman filter (+ Extended Kalman filter)

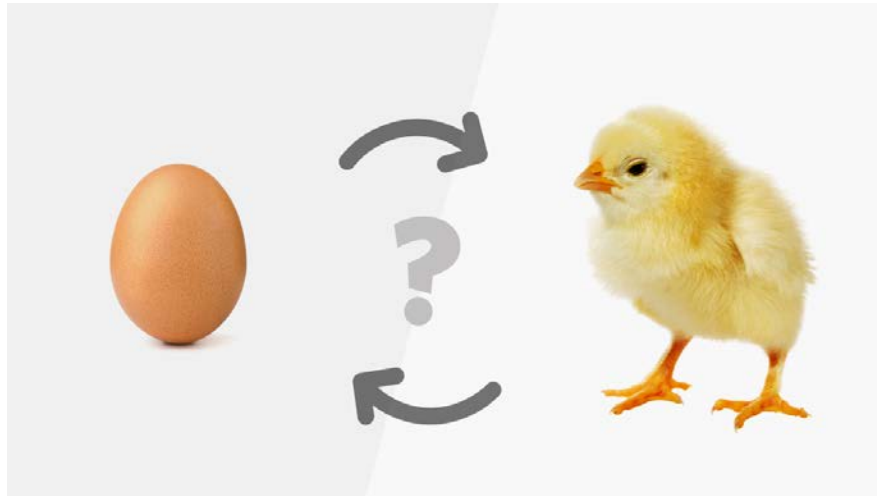


It's a weighted mean!

SLAM의 어려움

14

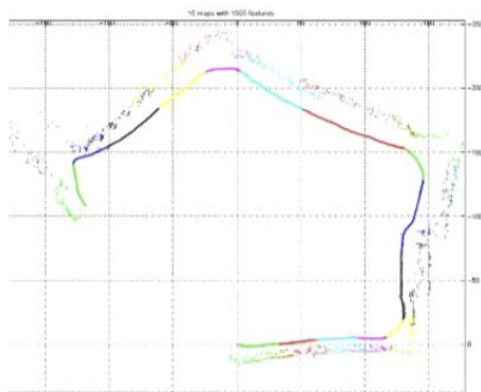
- SLAM이 어려운 문제인 이유
 - ▣ 기존의 단일 Localization / Mapping 기법들로 해결 불가능
 - Localization을 위해 (내 위치를 찾기 위해) 주변 지형의 위치를 알아야함
 - Monte-Carlo Localisation, Kalman Filter
 - Mapping을 위해 (주변 지형의 위치를 알기 위해) 내 위치를 알아야함
 - ▣ Localization이 먼저인가? Mapping이 먼저인가?



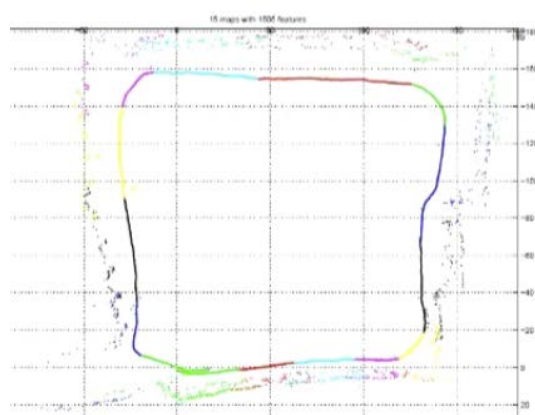
SLAM의 어려움

15

- SLAM이 어려운 문제인 이유 (계속)
 - ▣ 완벽한 state-estimation이 나올 수 없음
 - Monte-Carlo localisation의 경우, sampling을 통한 particle selection이기 때문에 optimal value를 얻을 수 없음
 - KF / EKF의 경우, 선형 근사를 하기 때문에 근사 에러가 생길 수 밖에 없음 (보통 비선형 데이터)
 - ▣ 시간이 지날수록 에러가 누적되며 추정된 global 위치 값에 큰 오차를 줌



Visual odometry



Visual SLAM

SLAM의 기술적 요소

16

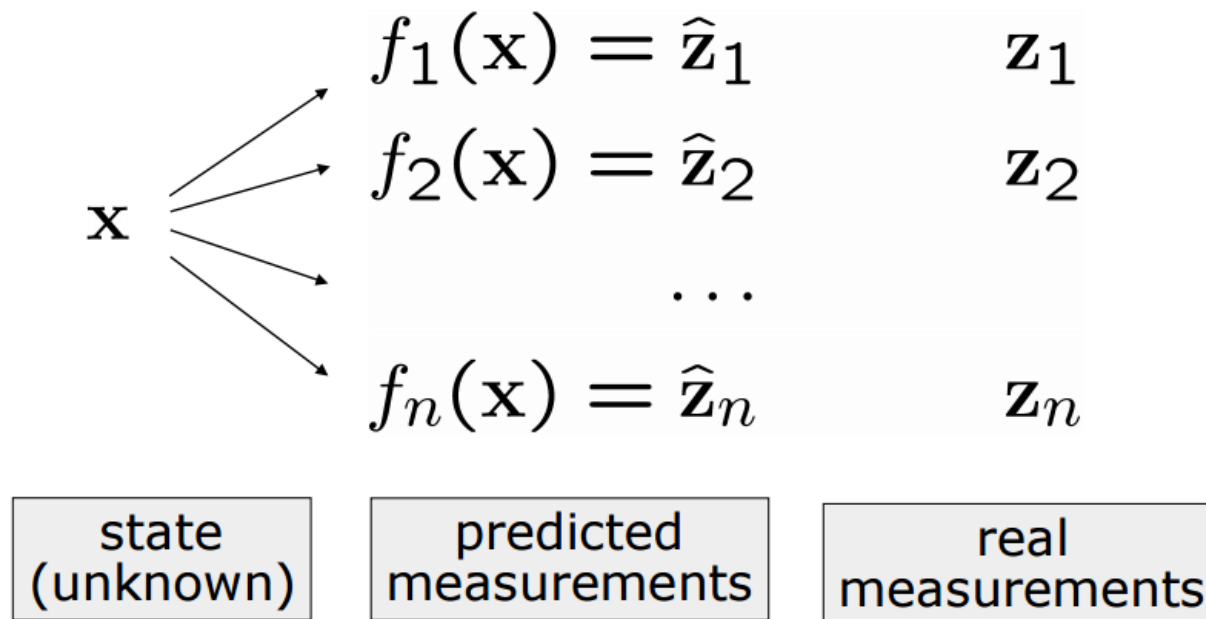
- SLAM은 이 문제를 어떻게 해결하나?
 - ▣ 연속된 센서 데이터로부터 가장 정확한 Localization 값과 가장 정확한 Mapping 값을 도출해낼 수 있는 최적화 문제로 (i.e. Least squares problem) 문제를 재정의함
 - ▣ Least-squares란?
 - Over-determined system의 해를 구하기 위한 방법
 - (거의 모든 상황의 SLAM 문제는 over-determined system임)
 - Over-determined system == 미지수의 수 보다 식의 수가 더 많은 경우
 - i.e., state의 수 보다 센서 값의 수가 더 많음
 - 실제 센서 값과 현재 state 값에서 예상되는 센서 값의 차이를 최소화

SLAM의 기술적 요소

17

□ Least-squares란?

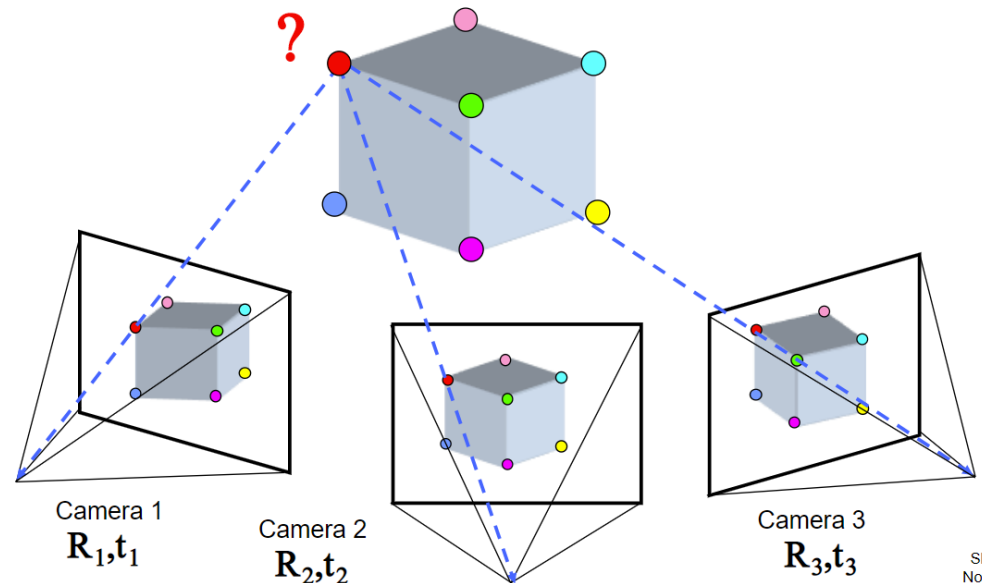
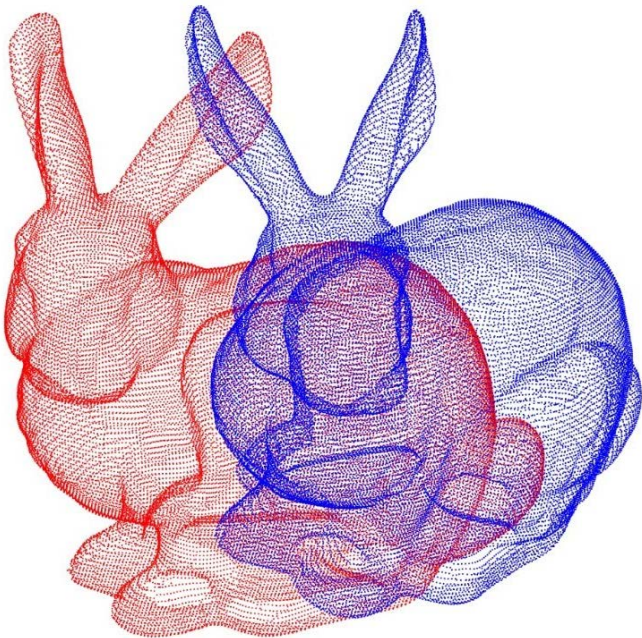
- ▣ 현재 로봇의 센서 값과 예상 센서 값을 최소화할 수 있는 로봇의 위치 + 맵의 위치 값은 무엇인가?



SLAM의 기술적 요소

18

- Least-squares란?
 - ▣ LiDAR SLAM - Iterative Closest Point (ICP)
 - ▣ Visual SLAM - Bundle Adjustment (BA)

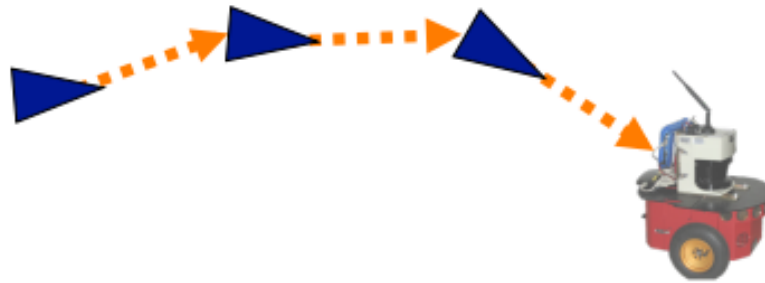


Slide credit:
Noah Snavely

SLAM의 기술적 요소

19

- Graph-based SLAM - Least-squares을 SLAM에 적용하는 방법
 - ▣ 로봇의 움직임과 위치를 Node와 Edge로 구성
 - ▣ 로봇이 odometry 정보로 움직이면서 motion constraint를 구성



Robot pose

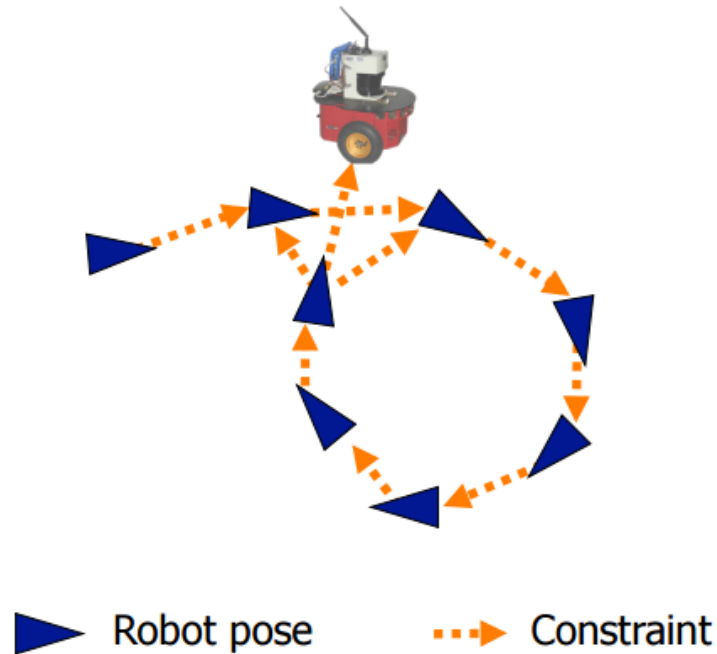


Constraint

SLAM의 기술적 요소

20

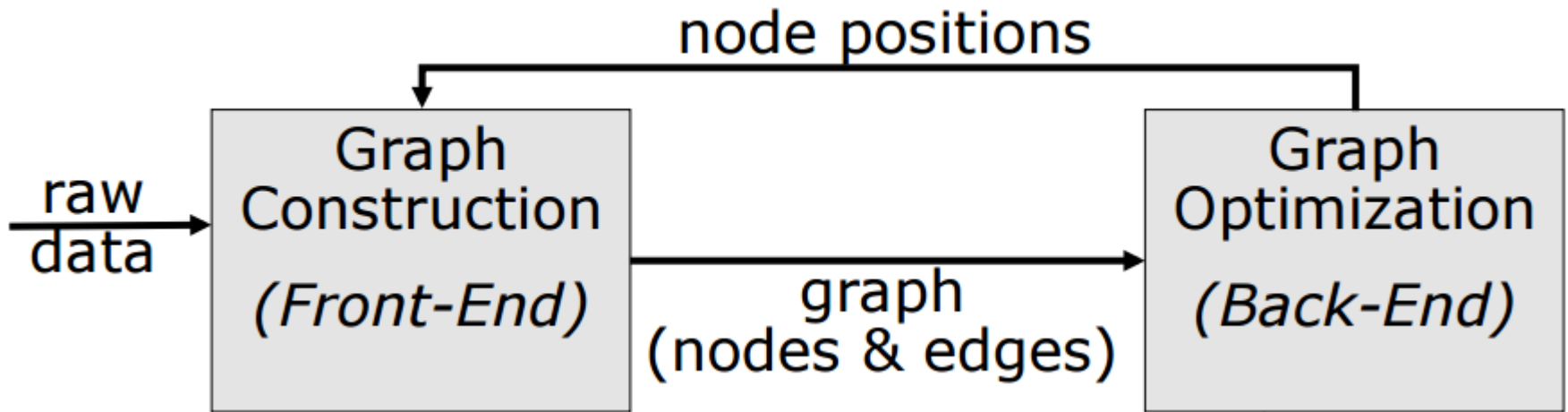
- Least-squares 문제를 구성하기 위해 Graph-based SLAM이 사용됨
 - ▣ 추가 constraint (e.g. loop closure) 등이 생성되면 최적화 수행 가능



SLAM의 기술적 요소

21

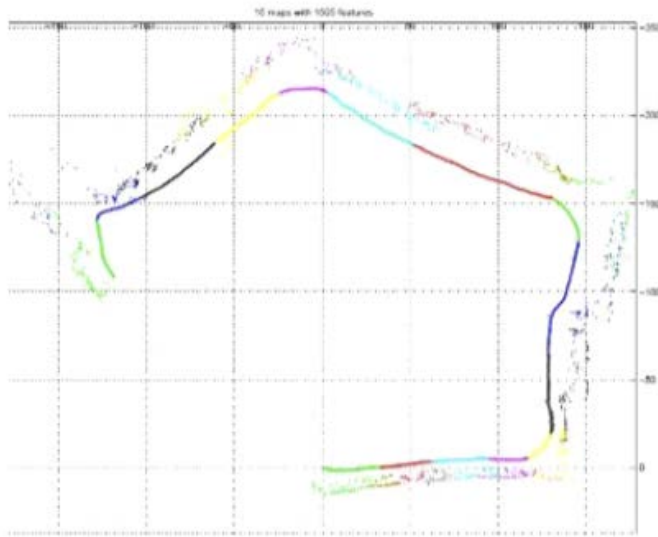
- Graph-SLAM의 프로그램 플로우
 - ▣ Front-end = 그래프 생성
 - ▣ Back-end = 그래프 최적화



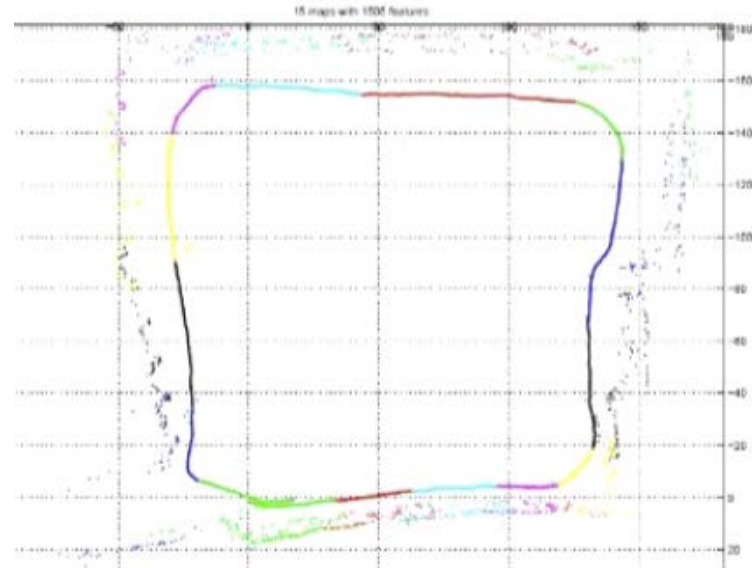
SLAM의 기술적 요소

22

- Graph-SLAM의 프로그램 플로우 (계속)
 - ▣ Global graph optimization이 없으면 odometry
 - ▣ Global graph optimization이 있으면 SLAM (비선형성, 다중구간 최적화를 고려)
 - Scaramuzza 의 SLAM 정의를 따름



Visual odometry



Visual SLAM

SLAM의 기술적 요소

23

- SLAM의 종류
 - ▣ Visual-SLAM (카메라 only)
 - ▣ LiDAR-SLAM (2D/3D LiDAR only)



SLAM 알고리즘의 종류

24

- 세부적 SLAM 기술 요소:
 - ▣ 다른 센서를 조합하는가?
 - Camera + IMU (Visual-Inertial Odometry)
 - LiDAR + IMU (LiDAR-Inertial Odometry)
 - Radar, GNSS... (...Sensor Fusion)
 - ▣ 여러 개의 센서를 융합 사용하는가?
 - Multi-camera SLAM
 - Collaborative SLAM
 - ▣ 특수 센서를 사용하는가?
 - Omnidirectional camera

SLAM의 이해

25

- SLAM에 대한 오해 1: Mapping 하는 데에 쓰기 좋음
 - ▣ Map 데이터도 least-squares에서 최적화하기는 함
 - ▣ 하지만 진짜로 정확한 Mapping이 필요하다면 실시간성 제한이 없는 Structure-from-Motion (SfM)을 사용하면 됨
- SLAM에 대한 오해 2: Data-driven 방식이 아니라 어디서든 잘됨
 - ▣ 환경이 바뀌면 파라미터 튜닝을 다시 해야할 수도 있음
 - ▣ 움직이는 물체가 있으면 안될 수 있고, 없어도 안될 수 있음
 - ▣ 딥러닝 기반이면 데이터에 의존성을 가질 수 있음
- SLAM에 대한 오해 3: CPU만 씬
 - ▣ 알고리즘에 따라 GPU를 써야하기도 함
 - Stereo visual-SLAM
 - RGB-D SLAM
 - LiDAR SLAM