Lidar Clustering

WeGo & WeCAR

WeGo

목 차

WeGo

01 Lidar Sensor

WeGo

01 Lidar Sensor

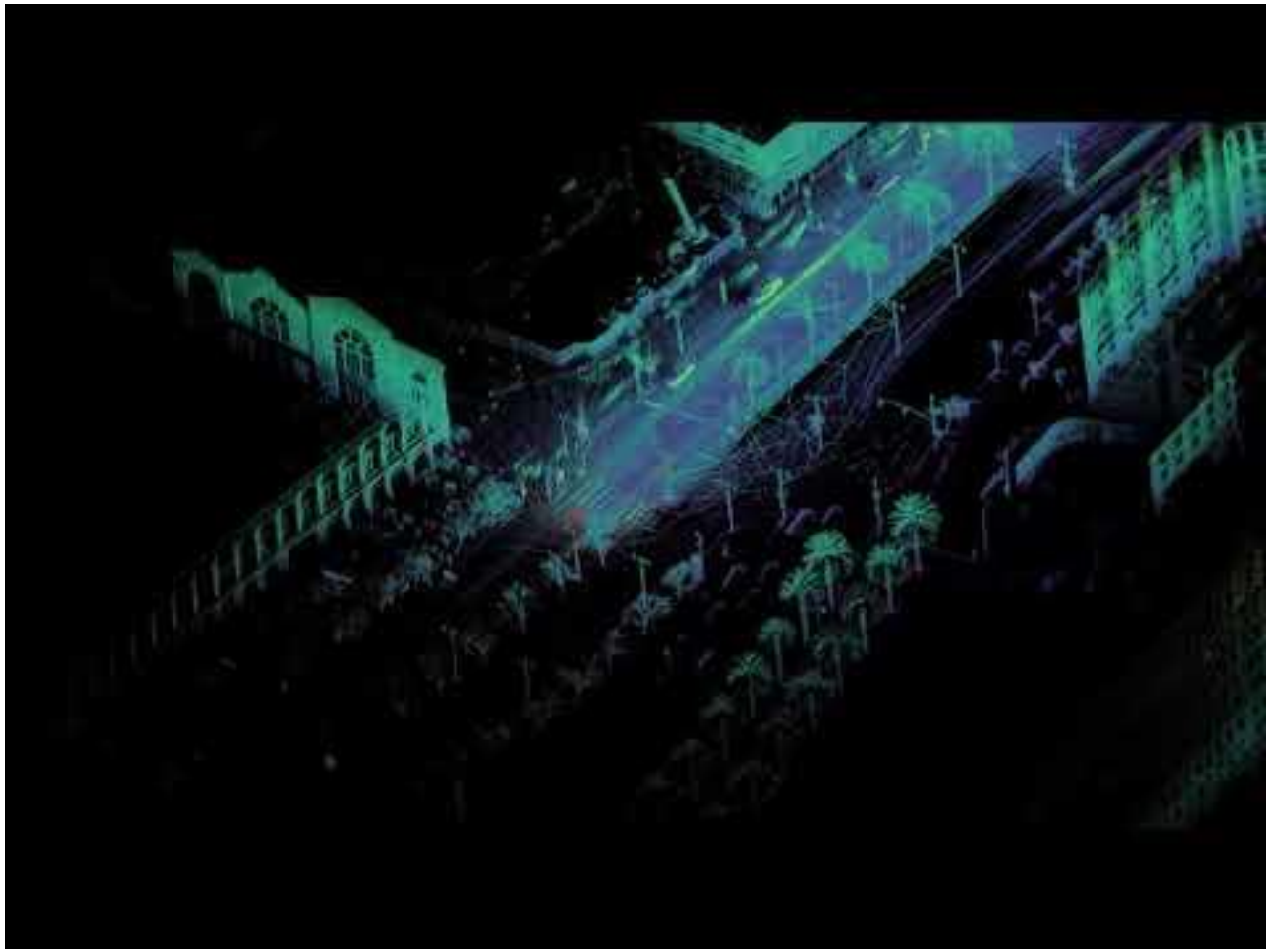
- LiDAR : Light Detection And Ranging
- 발사된 레이저(빛)가 물체에서 반사되어 돌아오는 시간을 이용하여 반사체와의 거리를 측정하는 센서
- 측정된 거리에 대한 신뢰도가 매우 높고, 분해능도 높은 편이다.
- 레이더에 비해, 파장이 짧으므로 공간 분해능이 높다.
- 다른 센서에 비해, FOV가 넓다
- 투과성이 없어서, 설치 공간의 제약이 있으며, 기상의 영향을 받는다.
- 빛을 사용하므로, 물체의 반사도에 따라 결과 값이 영향을 받게 된다.

WeGo

01 Lidar Sensor

- 3D Lidar based Mapping

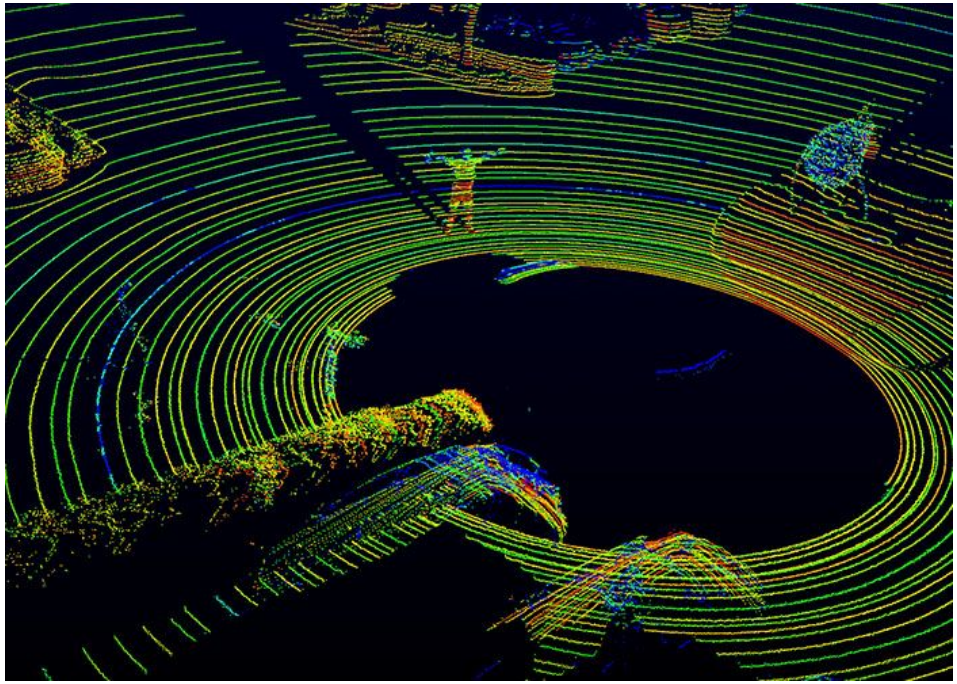
  http://www.youtube.com/watch?v=otp63J0ve2s



WeGo

01 Lidar Sensor

- Lidar의 특징 중 하나는 공간 해상도가 높다는 것
- 공간에 있는 모든 물체에 대해 측정이 진행됨
- 원하지 않는 물체 (바닥) 에 대해서도 측정이 진행됨 → 지면을 제거해야함
- 하나의 물체에서 다양한 측정 값이 발생 → 물체 별로 점들을 점군으로 묶어야 함



WeGo

02 ROS Lidar Sensor Data

WeGo

02 ROS Lidar Sensor Data

- sensor_msgs/LaserScan.msg → 2D Lidar Data 표현에 사용
  - Header header
  - float32 angle_min
  - float32 angle_max
  - float32 angle_increment
  - float32 time_increment
  - float32 scan_time
  - float32 range_min
  - float32 range_max
  - float32[] ranges
  - float32[] intensities

02 ROS Lidar Sensor Data

- sensor_msgs/PointCloud.msg → 2D, 3D 모두 사용 가능, 점 개수가 적을 경우 사용
  - Header header
  - geometry_msgs/Point32[] points
  - ChannelFloat32[] channels

02 ROS Lidar Sensor Data

- sensor_msgs/PointCloud2.msg → 3D Lidar Data를 표현 시 사용
  - Header header
  - uint32 height
  - uint32 width
  - PointField[] fields
  - bool is_bigendian
  - uint32 point_step
  - uint32 row_step
  - uint8[] data
  - bool is_dense

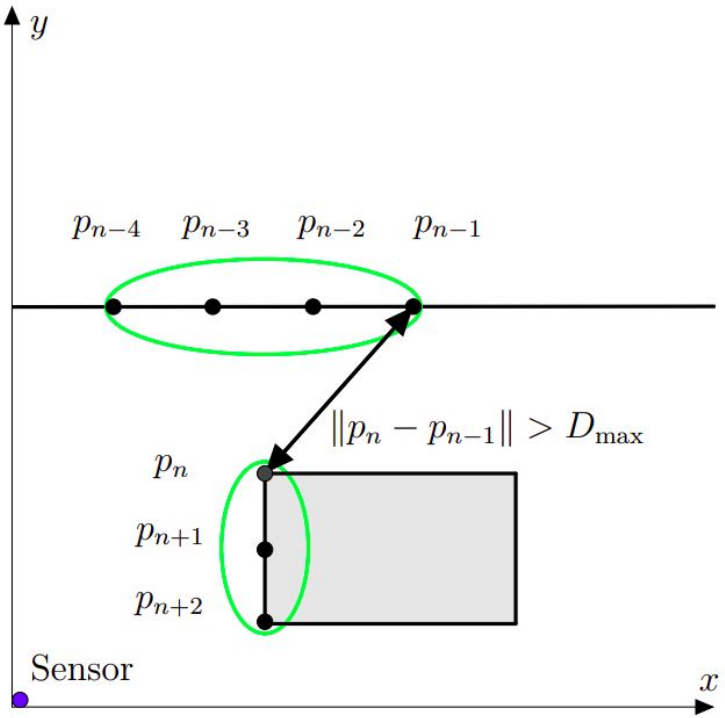
03 Lidar Clustering Algorithm

WeGo

03 Lidar Clustering Algorithm

- Lidar Clustering - Breakpoint
  - Lidar Clustering은 Lidar Point를 의미있는 결과로 묶는 것
  - 3D Lidar Data는 원하지 않는 데이터도 포함되어 있으므로, 제거가 필요 (지면 제거)
  - 2D Lidar Data는 전체 데이터를 이용하여 이를 특정 조건으로 묶어야함
  - 가장 간단한 방법은 거리가 가까운 점들을 하나의 Segment로 묶는 방법
- Breakpoint Detector

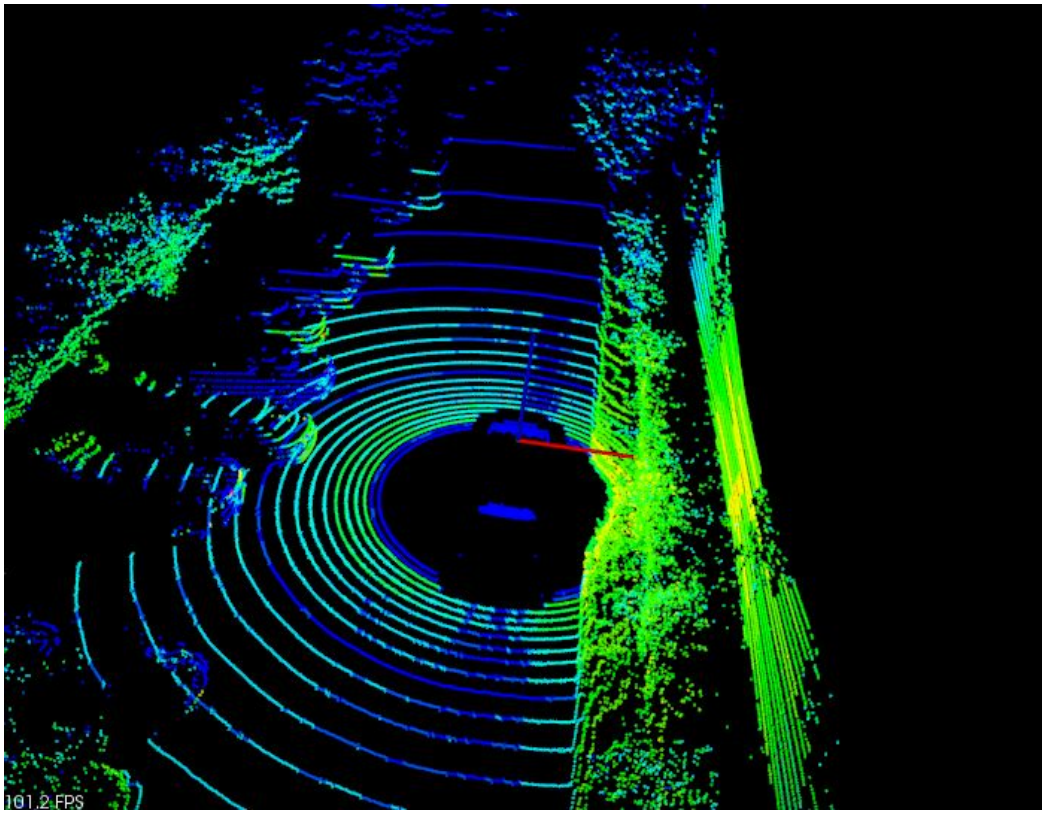
  ||Pn - Pn-1|| 〉 Dmax



WeGo

03 Lidar Clustering Algorithm

- Lidar Clustering - Breakpoint
  - Breakpoint Detector의 문제 → 라이다의 특성을 고려하지 않음
  - Lidar 데이터에서 근거리 물체의 점의 밀도와 원거리 물체의 점의 밀도가 다르다는 점
  - 공간 해상도가 다르므로. 센서 원점과의 거리를 고려해야 함



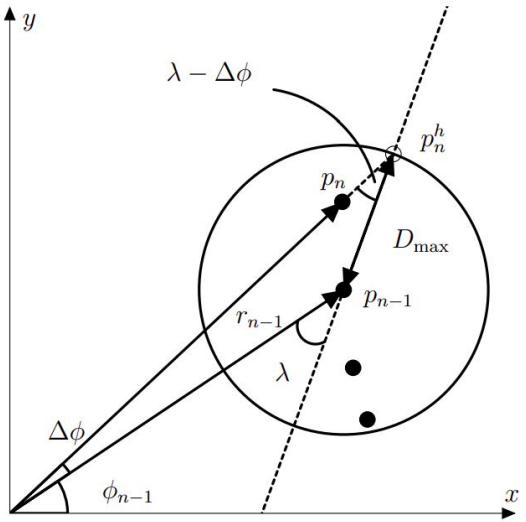
WeGo

03 Lidar Clustering Algorithm

- Lidar Clustering - Adaptive Breakpoint Detector
  - 멀리 있는 물체 → 점과 점들 사이의 거리가 멀어짐
  - 가까이 있는 물체 → 점과 점들 사이의 거리가 가까움
  - BreakPoint를 검출하는 Threshold 값을 측정값의 거리에 비례하게 설정
  - → Adaptive Breakpoint Detector

    (λ = Parameter, Δϕ = 각도 해상도, σ = 거리 해상도)
  - 단점 → 파라미터가 각도에 해당하므로, 직관적인 부분이 부족

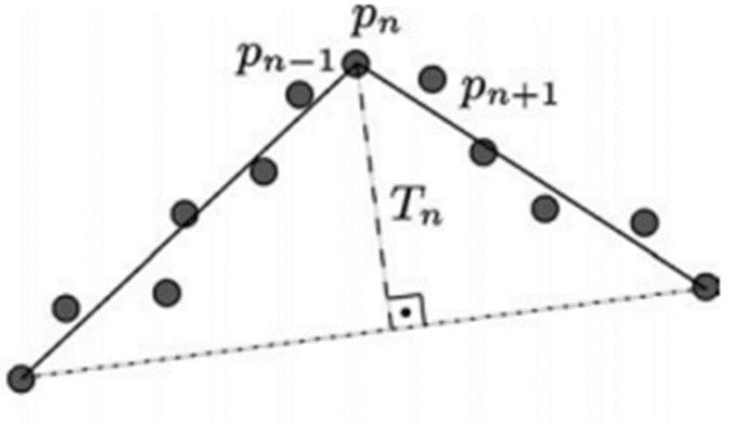
$$D_o = d_i \cdot \frac{\sin(\Delta\phi)}{\sin(\lambda - \Delta\phi)} + 3\sigma_d$$



WeGo

03 Lidar Clustering Algorithm

- Lidar Clustering - Split-and-Merge
  - Cluster 후, 의미있는 선으로 분해할 필요가 있음
  - 선 검출을 위해, Split이 필요
  - Cluster의 첫점과 끝점을 연결한 선과 최대 거리가 되는 점을 검출하고, 그 선과 점 사이의 거리가 일정 이상 일 경우, 분할, 아닐 경우 유지하는 방법을 통해 Split 진행



1. Cluster의 시작점과 끝점을 잇는 직선의 식을 계산
2. Cluster 내부의 각 점과 직선 사이의 거리 계산
3. 거리가 최대가 되는 점 Pn과 그 거리 Tn을 계산
4. Tn의 값이 임계값보다 작은 경우 다음 Cluster 진행
5. Tn의 값이 임계값보다 큰 경우, Pn을 기준으로 Cluster를 두 개로 분해

WeGo

04 Lidar Clustering Package

04 Lidar Clustering Package

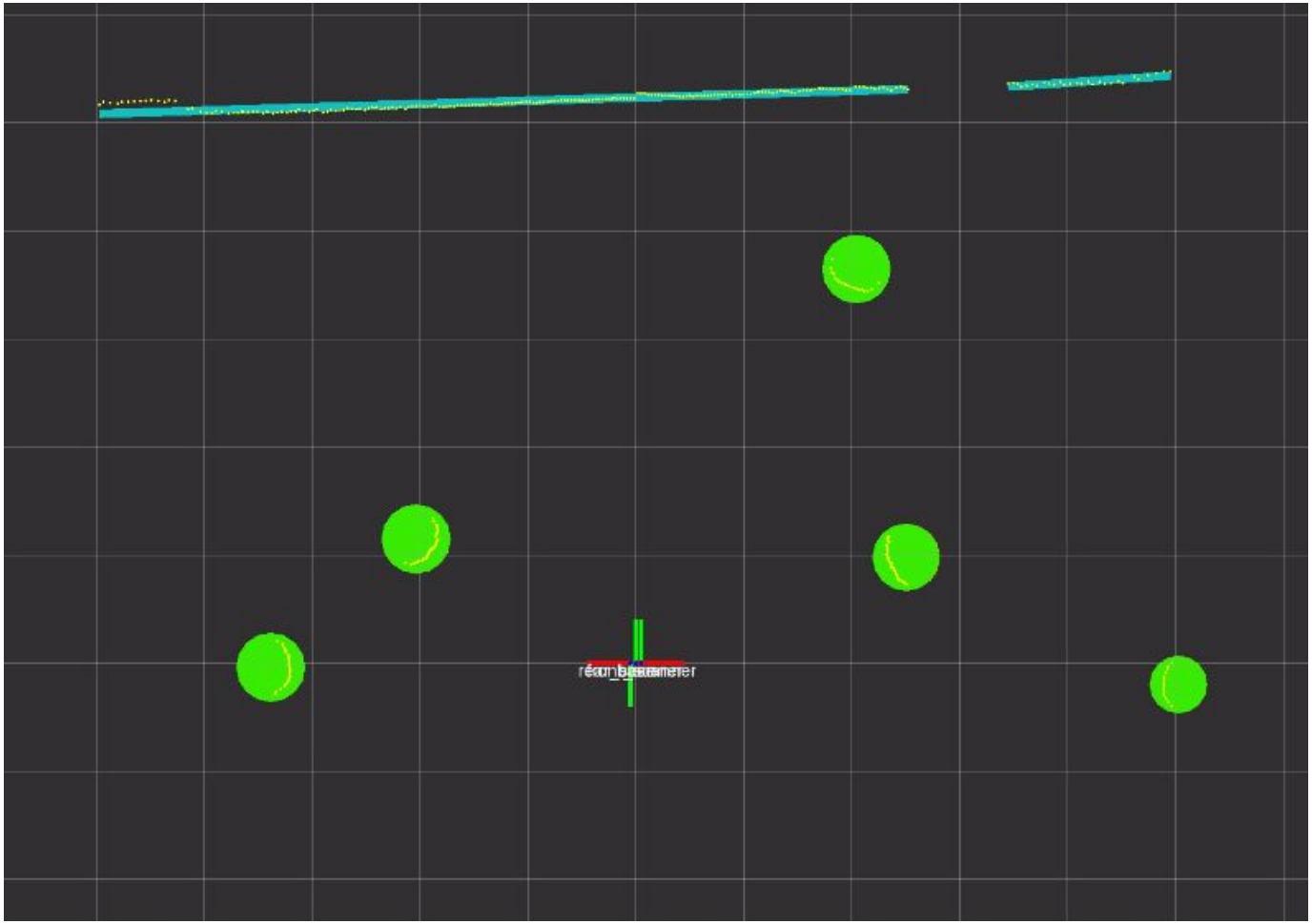
- Obstacle Detector Package
  - github - https://github.com/tysik/obstacle_detector
- Subscriber Topics
  - /scan (sensor_msgs/LaserScan)
- Published Topics
  - /obstacles (obstacle_detector/Obstacles)

04 Lidar Clustering Package

- Obstacle Detector Package
- Parameters
  - ~active (bool, default: true) – active/sleep mode.
  - ~use_scan (bool, default: false) – use laser scan messages.
  - ~use_pcl (bool, default: true) – use point cloud messages (if both scan and pcl are chosen, scans will have priority).
  - ~use_split_and_merge (bool, default: true) – choose wether to use Iterative End Point Fit (false) or Split And Merge (true) algorithm to detect segments.
  - ~circles_from_visibles (bool, default: true) – detect circular obstacles only from fully visible (not occluded) segments.
  - ~discard_converted_segments (bool, default: true) – do not publish segments, from which the circles were spawned.
  - ~transform_coordinates (bool, default: true) – transform the coordinates of obstacles to a frame described with frame_id parameter.
  - ~min_group_points (int, default: 5) – minimum number of points comprising a group to be further processed.
  - ~max_group_distance (double, default: 0.1) – if the distance between two points is greater than this value, start a new group.
  - ~distance_proportion (double, default: 0.00628) – enlarge the allowable distance between points proportionally to the range of point (use scan angle increment in radians).
  - ~max_split_distance (double, default: 0.2) – if a point in group lays further from a leading line than this value, split the group.
  - ~max_merge_separation (double, default: 0.2) – if distance between obstacles is smaller than this value, consider merging them.
  - ~max_merge_spread (double, default: 0.2) – merge two segments if all of their extreme points lay closer to the leading line than this value.
  - ~max_circle_radius (double, default: 0.6) – if a circle would have greater radius than this value, skip it.
  - ~radius_enlargement (double, default: 0.25) – artificially enlarge the circles radius by this value.
  - ~frame_id (string, default: map) – name of the coordinate frame used as origin for produced obstacles (used only if transform_coordinates flag is set to true).

04 Lidar Clustering Package

- Obstacle Detector Package
  - $ roslaunch obstacle_detector nodes.launch



WeGo

WeGo Robotics

Tel. 031 – 229 – 3553

Fax. 031 – 229 – 3554

제품 문의: go.sales@wego-robotics.com

기술 문의: go.support@wego-robotics.com