공학석사학위논문

다중 보행자 인지를 위한 센서 융합 알고리즘 개발

Sensor Fusion Algorithm for Multi Pedestrians Tracking

2019 년 8 월

서울대학교 대학원

기계항공공학부

문종식

Abstract

Sensor Fusion Algorithm

for Multi Pedestrians Tracking

Jongsik Moon

Mechanical and Aerospace Engineering

The Graduate School

Seoul National University

Pedestrian detection and tracking algorithm using environmental sensors is one of the most fundamental technology for safe urban autonomous driving. This paper presents a novel sensor fusion algorithm for multi pedestrian tracking using commercial vision sensor, LiDAR sensor, and digital HD map. The commercial vision sensor effectively detects pedestrian, whereas LiDAR sensor accurately measures a distance. Our system uses commercial vision sensor as detector and utilize LiDAR sensor to enhance estimation. In addition, digital HD map is utilized to properly define Region of Interest (ROI) of LiDAR sensor point cloud data. The detection performance is validated by about 4600 frames of SNU campus driving data and estimation accuracy is calculated through driving experiment. The proposed algorithm can be utilized for autonomous driving vehicles in various urban driving situation

Key Words: Autonomous Vehicle / Pedestrian / Commercial Vision Sensor / 2D LiDAR / HD Map / Sensor Fusion / Multi Target Tracking / Track to Track Fusion

Student Number: 2017-21074

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1. 1 Motivation

In a situation where technology level of ADAS (Active Driver Assistance System) is rising, the demand of technology is turning into full self-driving technology, which gives control to vehicles for safer driving than humans do. Social demands have also been strengthened. ADAS technology has previously been required to respond to scenarios in preparation for the behavior of forward vehicles which is important in highway driving scenario. However, the technology is now additionally required to cope with pedestrian scenarios. To prepare for this situation, the technology to clearly detect and track pedestrians must be implemented.

In order to cope with this technical trend, various companies have begun selling some modules with sensors which process data to detect objects in complex urban driving environments. Mobileye vision sensor, which processes vision data and hands over the state of object with classification label is a representative sensor for self-driving technology. In addition, Ibeo ECU uses point cloud data to do same operation.

However, these modules are also inherently adhering to the limits of the sensor. Classification performance limits exist for LiDAR sensor, especially for 2D Lidar sensor, with additional limits due to small vertical FOV. For vision sensor, accuracy limits exist for estimation performance for states such as position, velocity, yaw angle. Therefore, fusion between different sensors is essential to address these issues [Wei 2018].

1. 2 Previous Researches

A number of studies have been introduced for the development of a sensor fusion algorithm for multi pedestrian detection and tracking. Labayrade et al. [Labayrade et al. 2005] presented a fusion algorithm between LiDAR sensor and stereo vision sensor. This algorithm detects the lane using stereo vision sensor and find object using LiDAR sensor. Cristiano et al. [Premebida et al. 2009] proposed a detection level sensor fusion algorithm for pedestrian detection system. Some features of LiDAR detection are utilized for classification and setting Region of Interest (ROI) for vision sensor. Hyunggi Cho et al. [Cho et al. 2014] also presented vision, LiDAR, radar sensor fusion algorithm for detection and tracking of moving object. Utilizing a classification performance of vision sensor, they found objects and associated them with LiDAR sensor and radar sensor data measurements. Garcia et al. [Chavez-Garcia 2015] advanced fusion algorithm between LiDAR sensor, vision sensor, and radar to frontal object tracking in detection level. LiDAR sensor detection sets up Region of Interest (ROI) for vision sensor, and classification algorithm works in that region. Fusion algorithm and Multi Target Tracking (MTT) algorithm is implemented for after classification.

1. 3 Contributions

The cited approaches track only moving objects to avoid tracking static obstacles and does not use commercial vision sensor which detects and measure the states of pedestrians. The cited approaches can be used effectively on highways or in relatively simple urban driving situations where a regularized road rule exists. However, there are some situations where driving judgements should be carried out through interaction rather than by a regular road rule. In order to implement safer urban self-driving technologies in described situations, there is a need to construct cognitive algorithms that take into account the characteristics of pedestrians who are free to move and stop. Therefore, this paper proposes a commercial vision sensor, 2D LiDAR sensor, and digital HD map fusion architecture for pedestrian tracking which can be operated in severe urban driving situations. Using commercial vision sensor, LiDAR sensor measurement does not have to erase static point cloud data in a region where vision sensor can guide the existence of pedestrian. In addition, the prior knowledge of pedestrian existence can be used for clustering, and filtering process of LiDAR track. This can lead an advancement for detection rate and estimation accuracy.

To validate fusion algorithm, the track initializing & association probability along longitudinal distance was calculated to properly set FOV of the algorithm. Next, detection rate of a vision track, association rate of each track was calculated based on driving experiment and SNU Campus driving data. Finally, the error of estimated x, y position states are analyzed to see the accuracy of the final state.

1. 4 Thesis Outline

Section 1 covers the vehicle platform and the entire algorithm architecture for the implementation of the algorithm. Section 3 and 4 will describe algorithms that track vision and lidar tracks, respectively, while Section 5 shows an explanation of how to associate and fuse the two tracks. The result of the actual data simulation of the algorithms are in Section 6 and are described in Section 7 as a final conclusion and a further complement.

Chapter 2

System Architecture

2. 1 Vehicle Sensor Configuration

The vehicles used in this study are Hyundai IONIQ vehicles which equipped with sensors for autonomous driving. The configuration is as shown in Fig.1 through 2.

The Commercial Vision sensor uses Mobileye sensor. This vision sensor detects pedestrian and measures the position of pedestrian in local coordinate. Nevertheless, due to the limits of sensor FOV and inaccuracy of velocity measurements, the data from this sensor requires an additional tracking algorithm. Meanwhile, the LiDAR sensor uses IBEO lux, which covers about 110 degrees horizontally and 6 degrees vertically. Using six LiDAR sensors, all the horizontal FOVs are covered, but not fully covered in vertical direction. Due to these characteristics, it is difficult to recognize pedestrians by LiDAR alone. In addition, the position, yaw angle, and velocity states of vehicle is measured by GPS, RT-3002.

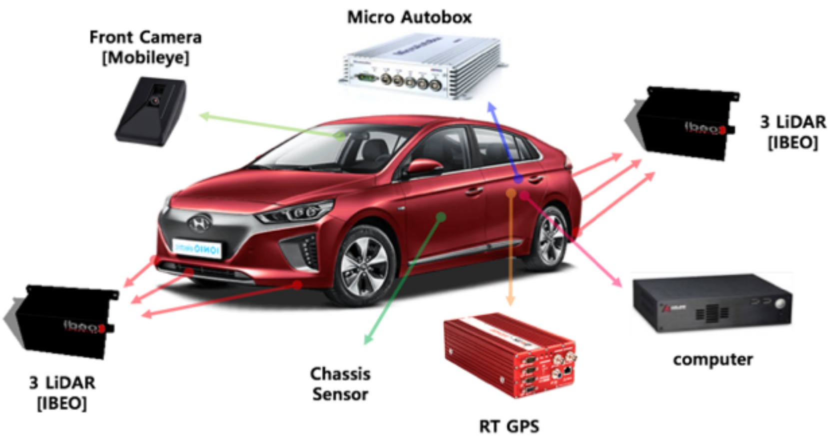

Fig. 1 Vehicle platform and sensor configuration

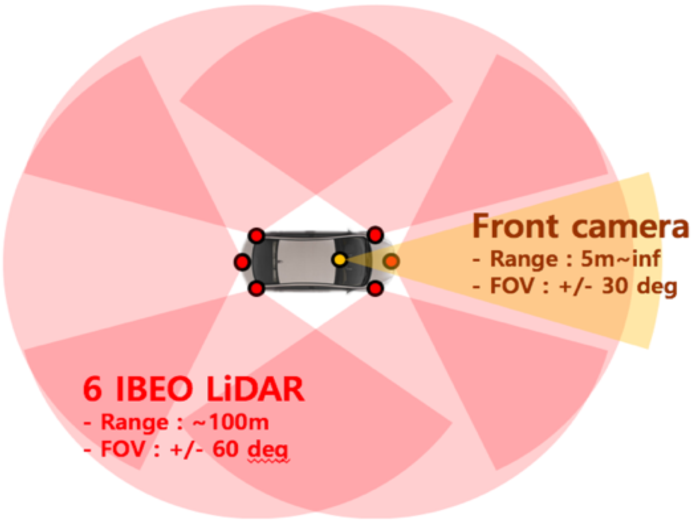

Fig. 2 Sensor Field of View (FOV)

2. 2 Fusion Architecture

Fig. 3 shows the schematic of our proposed sensor fusion architecture. The algorithm receives position information & classification information from vision sensor, digital maps, and point cloud data from LiDAR sensor. In order to utilize an accurate classification performance of vision sensor, vision-only track is initialized and sets up a ROI for LiDAR sensor. Using the point cloud in the ROI, LiDAR sensor detection and tracking algorithm operates with proper parameter for pedestrian detection and tracking. The final two states are associated with GNN (Global nearest neighbor) algorithm [Baig 2012, Blackman 2004] and fused using Covariance Intersection (CI) algorithm [Julier and Uhlmann 2009]. At this time, 2D LiDAR sensor uses only the data inside the road by setting an additional ROI based on the digital HD Map, since it is difficult to use data outside the road, especially on the sidewalk. Vision track, which is not fused, can be tracked with vision-only measurement to enhance robustness of algorithm. Based on this, states of pedestrian can be tracked so that planning algorithms for autonomous driving can operate.

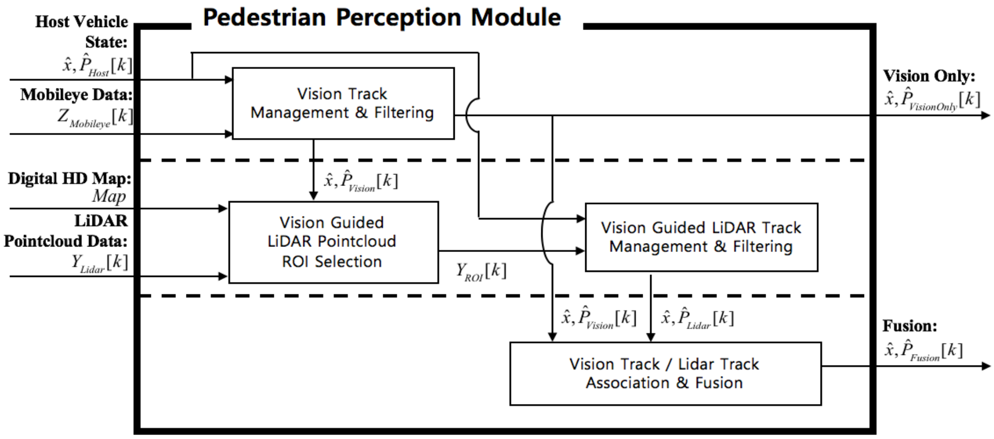

Fig. 3 Fusion system architecture for pedestrian detection and tracking

Chapter 3

Vision Track Management & Filtering

Based on the MTT method, the position data of the commercial vision sensor are used to estimate the status of pedestrians. The overall schematic diagram is as shown in Fig.4 below.

The MTT algorithm refers to the process of estimating a state through filtering while managing the track by creating possible tracks based on measurement values and continuously assigning them measurement [Bar-Shalom 1990, Blackman et al. 1999, Thrun 2005]. For vision sensor tracking algorithm, Extended Kalman Filter (EKF) is used as filtering method, and GNN algorithm is utilized as data association algorithm required for track management.

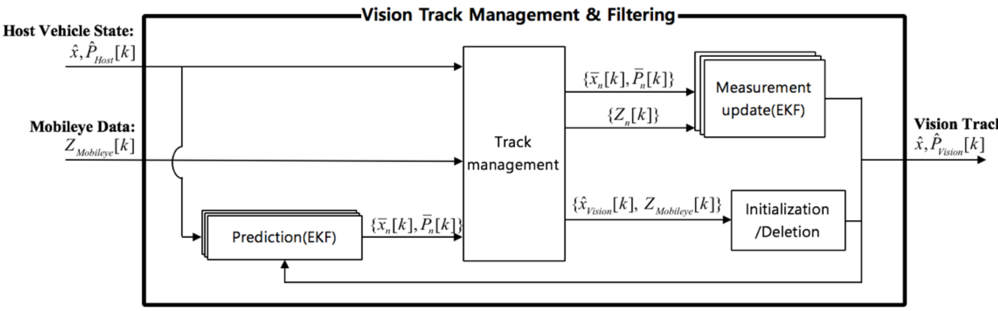

Fig. 4 Vision Track Management & Filtering architecture

3. 1 Filtering for Target Tracking

The states of each track are estimated by EKF algorithm. The process model and measurement are described in 3.1.1, 3.1.2 respectively.

3. 1. 1 Process Model

In this paper, Kim's proposed process model is utilized [Kim et al. 2014]. The continuous process model is as follows.

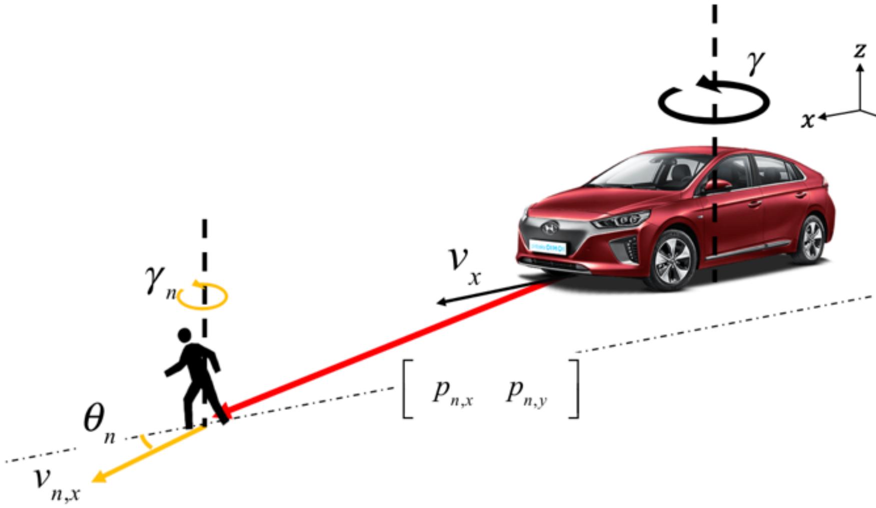

Fig. 5 Physical meaning of each variable

In order to describe all the motions, the state and input vector are defined as follows:

$$\mathbf{x}_{\mathrm{n}} = \begin{bmatrix} p_{n,x} & p_{n,y} & \theta_n & v_{n,x} & \gamma_n & a_n & \dot{\gamma}_n \end{bmatrix}^T$$
$$\mathbf{u} = [v_x \quad \gamma]^T \tag{3.1}$$

where subscript x and y denotes longitudinal, and lateral axis of vehicle coordinate. Another subscript n denotes n-th target. p denotes a relative position. θ denotes relative yaw angle. γ denotes yaw angle rate, a denotes acceleration. These parameters are described in Fig. 5.

Using the state and input vector above, process model can be defined as

$$
\begin{aligned}
\dot{\mathbf{x}}_n &= \mathbf{a}(\mathbf{x}_n, \mathbf{u}) + \mathbf{q} \\
&= [\mathbf{a}_1 \quad \mathbf{a}_2 \quad \mathbf{a}_3 \quad \mathbf{a}_4 \quad \mathbf{a}_5 \quad \mathbf{a}_6 \quad \mathbf{a}_7]^T + \mathbf{q} \\
\mathbf{a}_1 &= v_{n,x} \cos\theta_i - v_x + p_{n,y} \cdot \gamma \\
\mathbf{a}_2 &= v_{n,x} \sin\theta_i - p_{n,x} \cdot \gamma \\
\mathbf{a}_3 &= \gamma_n - \gamma \\
\mathbf{a}_4 &= a_{n,x} \\
\mathbf{a}_5 &= \dot{\gamma}_n \\
\mathbf{a}_6 &= -k_a \\
\mathbf{a}_7 &= -k_{\dot{\gamma}}
\end{aligned}
\tag{3.2}
$$

where k_a denotes a decay rate of acceleration and $k_{\dot{\gamma}}$ denotes a decay rate of yaw acceleration. Above continuous process model can be discretized via Taylor methods as follows [Kazantzis'99]:

$$
\begin{aligned}
\mathbf{x}_n[k+1] &= \mathbf{x}_n(t + \Delta T) \\
&= \mathbf{x}_n(t) + \frac{d}{dt}\mathbf{x}_n \cdot \Delta T + \frac{1}{2}\frac{d^2}{dt^2}\mathbf{x}_n \cdot \Delta T^2 + \mathbf{h.o.t.} \\
&\cong \mathbf{x}_n(t) + \{\mathbf{a}(\mathbf{x}_n, \mathbf{u}) + \mathbf{q}\} \cdot \Delta T + \frac{1}{2}\frac{d}{dt}\{\mathbf{a}(\mathbf{x}_n, \mathbf{u}) + \mathbf{q}\} \cdot \Delta T^2
\end{aligned}
\tag{3.3}
$$

$$= \left[\mathbf{x}_n(t) + \{\mathbf{a}(\mathbf{x}_n, \mathbf{u}) + \mathbf{q}\} \cdot \Delta T + \frac{1}{2}\left\{\frac{\partial}{\partial \mathbf{x}_n}\mathbf{a}(\mathbf{x}_n, \mathbf{u}) \cdot \dot{x_n} + \frac{\partial}{\partial \mathbf{u}}\mathbf{a}(\mathbf{x}_n, \mathbf{u}) \cdot \dot{\mathbf{u}}\right\} \cdot \Delta T^2\right]$$

$$= \Bigg[\mathbf{x}_n(t) + \mathbf{a}(\mathbf{x}_n, \mathbf{u}) \cdot \Delta T + \frac{1}{2}\left\{\frac{\partial}{\partial \mathbf{x}_n}\mathbf{a}(\mathbf{x}_n, \mathbf{u}) \cdot \dot{x_n} + \frac{\partial}{\partial \mathbf{u}}\mathbf{a}(\mathbf{x}_n, \mathbf{u}) \cdot \dot{\mathbf{u}}\right\} \cdot \Delta T^2$$

$$+ \left\{\Delta T \cdot \mathbf{I} + \frac{\Delta T^2}{2} \cdot \frac{\partial}{\partial \mathbf{x}_n}\mathbf{a}(\mathbf{x}_n, \mathbf{u})\right\}\mathbf{q}\Bigg]$$

$$= \mathbf{f}(\mathbf{x}_n[k], \mathbf{u}[k]) + \mathbf{w}[k]$$

$$= [\mathbf{f}_1 \quad \mathbf{f}_2 \quad \mathbf{f}_3 \quad \mathbf{f}_4 \quad \mathbf{f}_5 \quad \mathbf{f}_6 \quad \mathbf{f}_7]^{\boldsymbol{T}} + \mathbf{w}[k]$$

where

$$\mathbf{f}_1 = -\begin{pmatrix} \dfrac{\gamma_{host}(p_{n,x}\gamma_{host} - v_n \sin(\theta_n))}{2} \\ -\dfrac{a_n \cos(\theta_n)}{2} + \dfrac{\gamma_n v_n \sin(\theta_n)}{2} \\ +\dfrac{a_{host}}{2} - \dfrac{p_{n,y}\dot{\gamma}_{host}}{2} \end{pmatrix} \Delta T^2 + \begin{pmatrix} p_{n,y}\gamma_{host} - v_{host} \\ +v_n \cos(\theta_n) \end{pmatrix} \Delta T + p_{n,x}$$

$$\mathbf{f}_2 = -\begin{pmatrix} \dfrac{a_n \sin(\theta_n)}{2} + \dfrac{\gamma_n v_n \cos(\theta_n)}{2} + \dfrac{p_{n,x}\dot{\gamma}_{host}}{2} \\ -\dfrac{\gamma_{host}(p_{n,y}\gamma_{host} - v_{host} + v_n \cos(\theta_n))}{2} \end{pmatrix} \Delta T^2 - \begin{pmatrix} p_{n,x}\gamma_{host} \\ -v_n \sin(\theta_n) \end{pmatrix} \Delta T + p_{n,y}$$

$$\mathbf{f}_3 = \frac{\dot{\gamma}_n}{2}\Delta T^2 + \gamma_n \Delta T + \theta_n$$

$$\mathbf{f}_4 = -\frac{a_n k_a}{2}\Delta T^2 + a_n \Delta T + v_n \tag{3.4}$$

$$\mathbf{f}_5 = -\frac{k_{\dot{\gamma}}\dot{\gamma}_n}{2}\Delta T^2 + \dot{\gamma}_n \Delta T + \gamma_n$$

$$\mathbf{f}_6 = \frac{a_n}{2}(k_a^2 \Delta T^2 - 2k_a \Delta T + 2)$$

$$\mathbf{f}_7 = \frac{\dot{\gamma}_n}{2}\left(k_{\dot{\gamma}}^2 \Delta T^2 - 2k_{\dot{\gamma}} \Delta T + 2\right)$$

$$\mathbf{w}[k] \sim (0, \mathbf{W}[k])$$

$$\mathbf{W}[k] = \boldsymbol{E}[\mathbf{w} \cdot \mathbf{w}^T]$$

$$= E\left[\left[\left\{\Delta T \cdot \mathbf{I} + \frac{\Delta T^2}{2} \cdot \frac{\partial}{\partial \mathbf{x}_n} \mathbf{a}(\mathbf{x}_n, \mathbf{u})\right\} \mathbf{q}\right] [\dots]^T\right]$$

$$= \left\{\Delta T \cdot \mathbf{I} + \frac{\Delta T^2}{2} \cdot \frac{\partial}{\partial \mathbf{x}_n} \mathbf{a}(\mathbf{x}_n, \mathbf{u})\right\} \mathbf{E}[\mathbf{q} \cdot \mathbf{q}^T]\{\dots\}^T$$

$$= \left\{\Delta T \cdot \mathbf{I} + \frac{\Delta T^2}{2} \cdot \frac{\partial}{\partial \mathbf{x}_n} \mathbf{a}(\mathbf{x}_n, \mathbf{u})\right\} \mathbf{Q}\{\dots\}^T \Bigg|_{\substack{\mathbf{x}_n = \mathbf{x}_n[k] \\ \mathbf{u} = \mathbf{u}[k]}}$$

$$\dot{\mathbf{u}} = \frac{d}{dt}([v_x \quad \gamma]^T) = [a_x \quad \dot{\gamma}]^T$$

3. 1. 2 Measurement Model

Commercial vision sensor detects pedestrian and measure a relative distance. Therefore, the measurement vector can be defined as follows:

$$\mathbf{z}_\mathrm{n} = \begin{bmatrix} p_{n,x} & p_{n,y} \end{bmatrix}^T \tag{3.5}$$

Using the state, measurement model can be defined as

$$\begin{gathered} \mathbf{z}_\mathrm{n}[k] = \mathbf{H}_n \mathbf{x}_n[k] + \mathbf{v}_n[k] \\ \mathbf{v}_n[k] \sim (0, \mathbf{V}_n[k]) \\ \mathbf{H}_n = \begin{bmatrix} 1\ 0\ 0\ 0\ 0\ 0\ 0 \\ 0\ 1\ 0\ 0\ 0\ 0\ 0 \end{bmatrix} \end{gathered} \tag{3.6}$$

$\mathbf{V}_n[k]$ is calculated based on driving experiment data.

3. 2 Data Association

One of the essential methodologies for track management is the data association algorithm. Various algorithms have been presented such as Joint Probability Data Association Filter (JPDAF), Multiple Hypothesis Tracking (MHT). In this paper, Global Nearest Neighborhood algorithm is utilized for vision-only tracking algorithm. Commercial vision sensor data has little noise. Therefore, a strong association algorithm for can be utilized without degrading a performance [Bar-Shalom 1990, Bar-Shalom 1995]. For GNN algorithm, a cost matrix $\boldsymbol{A} = [\boldsymbol{c}_{ij}]$ for associating track to measurement should be defined. We defined the cost for associating as

$$\begin{aligned} \boldsymbol{c}_{ij} &= \boldsymbol{\zeta}_{\mathrm{ij}}[k]^T \cdot \mathbf{W} \cdot \boldsymbol{\zeta}_{ij}[k] \\ \boldsymbol{\zeta}_{\mathrm{ij}}[k] &= \mathbf{z}_{\mathrm{i}}[k] - \mathbf{H}_j \mathbf{x}_j^-[k] = [\Delta\mathbf{x} \quad \Delta\mathbf{y}] \end{aligned} \tag{3.7}$$

which denotes 2-norm position difference which is called as Mahalanobis Distance (MD).

Using GNN algorithm, the detected measurement is assigned to the track, or initialize a new track. In this paper, an initialized track lasts about 1 second after final data assignment.

Chapter 4

Vision Guided LiDAR Track Management & Filtering

LiDAR sensor measures the surrounding environment in the form of a point cloud. This type of data is difficult to handle individually and must be clustered. Using the center of cluster as position data, MTT algorithm is implemented for LiDAR track [Blackman 2004].

Since 2D LiDAR sensor is not efficient sensor for classification, the detected data is noisy. Given this situation, it can be determined that the Multiple Hypothesis Tracking (MHT) method, which produces possible hypothetical tracks based on

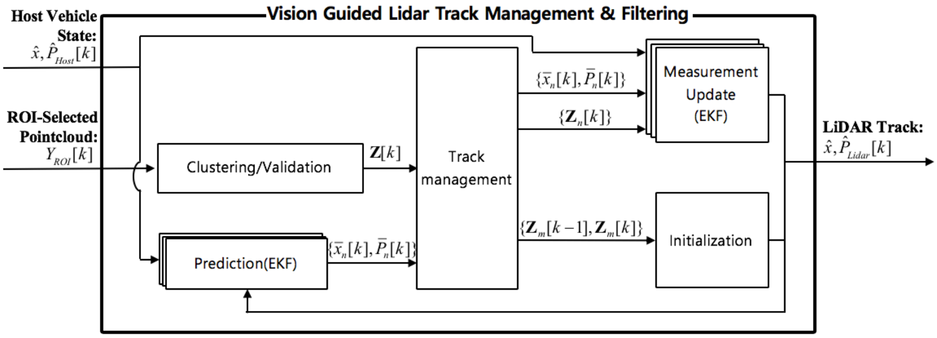

Fig. 6 Vision Guided LiDAR Track Management & Filtering architecture

measurements, continuously compares them with measurements, and uses only reliable tracks with increased reliability, is suitable for tracking. In addition, EKF algorithm also implemented for filtering process of MHT. The overall schematic is as shown in Fig.6.

4. 1 Cluster Validation

Before using MHT algorithm for tracking, clustering algorithm must operate. In this process, the feature of the cluster allows for simple verification of whether it is a suitable cluster for pedestrians. Three features used for validation are stated in Table 1.

Table 1. Features for cluster-validation

Feature	Condition
The number of points	$2 < n < 150$
Maximum eigenvalue	$0.001 < \lambda < 0.5$
Eigenvalue difference between max and min	$\Delta\lambda < 0.1$

4. 2 Filtering for Target Tracking

The process model and measurement model are described in 4.2.1 and 4.2.2 respectively.

4. 2. 1 Process Model

Vision track and LiDAR track utilize same process model which is presented by Kim et al at 3.1.1.

4. 2. 2 Measurement Model

Measurement model for LiDAR track utilizes additional measurements: heading angle and 2-norm velocity. Therefore, the measurement vector can be defined as follows:

$$\mathbf{z}_{\mathrm{n}} = \begin{bmatrix} p_{n,x} & p_{n,y} & \theta_n & v_{n,x} \end{bmatrix}^T \tag{4.1}$$

Using the state, measurement model can be defined as

$$\begin{gathered} \mathbf{z}_{\mathrm{n}}[k] = \mathbf{H}_n \mathbf{x}_n[k] + \mathbf{v}_n[k] \\ \mathbf{v}_n[k] \sim (0, \mathbf{V}_n[k]) \\ \mathbf{H}_n = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \end{gathered} \tag{4.2}$$

4. 3 Track Management Rule

LiDAR track management is implemented using MHT algorithm. Measurements are assigned to a track that is previously tracked and within a certain distance. Unassociated measurements initiate a new track, and unassociated tracks can be deleted if no measurements are assigned to the track more than three consecutive times or if no measurements are assigned for more than 30% of the time of track existence.

Chapter 5

Fusion Method

5. 1 Track Association

Track states are not noisy neither. Therefore, GNN algorithm which is a strong association method can be utilized. The cost for cost matrix $\boldsymbol{A} = [\boldsymbol{c}_{ij}]$ is defined as follows:

$$\boldsymbol{c}_{ij} = \boldsymbol{\zeta}_{\mathrm{ij}}[k]^T \cdot \mathbf{W} \cdot \boldsymbol{\zeta}_{ij}[k]$$
$$\boldsymbol{\zeta}_{\mathrm{ij}}[k] = \frac{1}{n}\sum_{t=1}^{n} \parallel p_{vision}[t] - p_{LiDAR}[t] \parallel_2 \tag{5.1}$$

where p denotes 2 by 1 relative position vector. In addition, to prevent vision tracks from being associated with tracks that are so far away, the cost $\boldsymbol{c}_{ij}$ has an upper limit.

5. 2 State Fusion

Since commercial vision sensor measurement and LiDAR sensor measurement have a correlation value that cannot be calculated because they process data using same vehicle information. This unknown correlation not only degrades the performance of fusion process, but also causes it to diverge in severe cases [Bar-Shalom 1981]. To this end, fusion of the two states is carried out using the algorithm of Covariance Intersection (CI), which can be used when the correlation value is not known. Using CI algorithm, the fusion state can be defined as follows:

$$\begin{gathered}
\mathbf{x}_{\mathrm{fus}} = \mathbf{P}_{\mathrm{fus}}(\omega\mathbf{P}_1^{-1}\mathbf{x}_1 + (1-\omega)\mathbf{P}_2^{-1}\mathbf{x}_2) \\
\mathbf{P}_{\mathrm{fus}=}(\omega\mathbf{P}_1^{-1} + (1-\omega)\mathbf{P}_2^{-1})^{-1} \\
\omega = \arg\min_{\omega\in[0,1]} tr\{(\omega\mathbf{P}_1^{-1} + (1-\omega)\mathbf{P}_2^{-1})^{-1}\}
\end{gathered} \tag{5.2}$$

Chapter 6

Experimental Result

To validate the proposed algorithm from the driver's point of view, we calculated three factors: track initializing & association probability along longitudinal distance, detection & association rate in the FOV, and error of estimated states.

Track initializing & association probability along longitudinal distance must be calculated to properly set the FOV and plan longitudinal control strategy. We validated the detection rate of LiDAR track and association rate for track fusion algorithm in the FOV using SNU campus driving data. Finally, an accuracy of estimated fusion is presented.

6. 1 Track Initializing and Association Probability along Longitudinal Distance

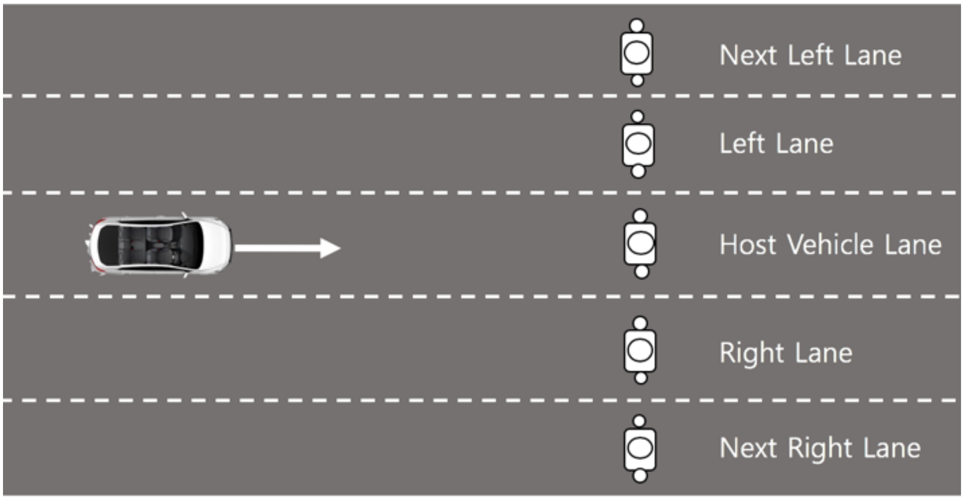

Fig. 7 Experiment for track initializing distance

To specify the FOV of vision sensor, a longitudinal distance of first sensing moment is required. Therefore, we set the pedestrian to standstill on the specific lane and moved vehicle toward the pedestrian to measure the first detected moment. This method is described in Fig. 7. We calculated the track initializing probability along longitudinal by repeating this method 12 times for five lateral distance. The experimental result for track initializing & association probability along longitudinal distance is described in Fig. 8.

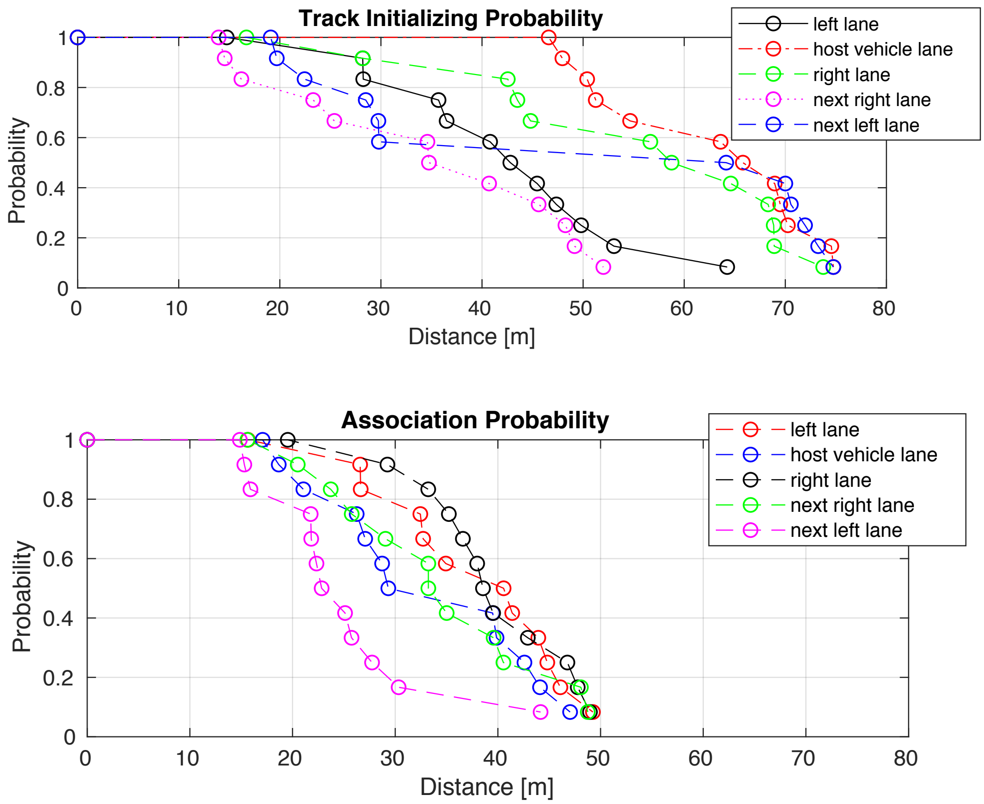

Fig. 8 Vision track initializing & two track association probability along longitudinal distance

X-axis denotes a longitudinal distance from host vehicle, and y-axis denotes a probability. Legends shows a lateral distance. As Fig. 8 denotes, track initializing probability where distance is more than 30m does not guarantee proper performance, which is less than 60%. Assuming that the detection performance has been reduced due to pedestrian being stopped, we set longitudinal distance of FOV less than 30m. In terms of association probability, the distance of 30m provides sufficient probability nearly more than 50 % for the three most important lanes for safe driving: host vehicle lane, right lane, left lane.

6. 2 Detection & Association Rate in SNU Campus Driving Data

In experiments 6.1, pedestrian stood still. Therefore, we additionally validated our algorithm with SNU campus driving data which can be more suitable to describe the actual movement of pedestrians. Simulation result is presented at Table 2.

The values in the table mean the number of frames. If pedestrian is in the FOV of vision sensor and is tracked using vision sensor measurements, the number is counted as detection, or non-detection if it does not be tracked. For LiDAR track, existence of pedestrian is guaranteed by vision track. Therefore, false positive cases are not counted. To summarize the result, precision, recall and accuracy are calculated in Table 2.

Table 2 Simulation result for vision track

Sensor Data	Existence	Detection	Non-Detection	Precision	Recall	Accuracy
Vision Sensor	Target Exist	460	34	100%	93.12%	99.27%
	Target Non-Exist	0	4151			
LiDAR Sensor	Target Exist	141	10	x	93.38%	x
		Associated	**Non-Associated**			
Vision & LiDAR	Two Tracks Exists	137	3	x	97.90%	x

6. 3 Error of States

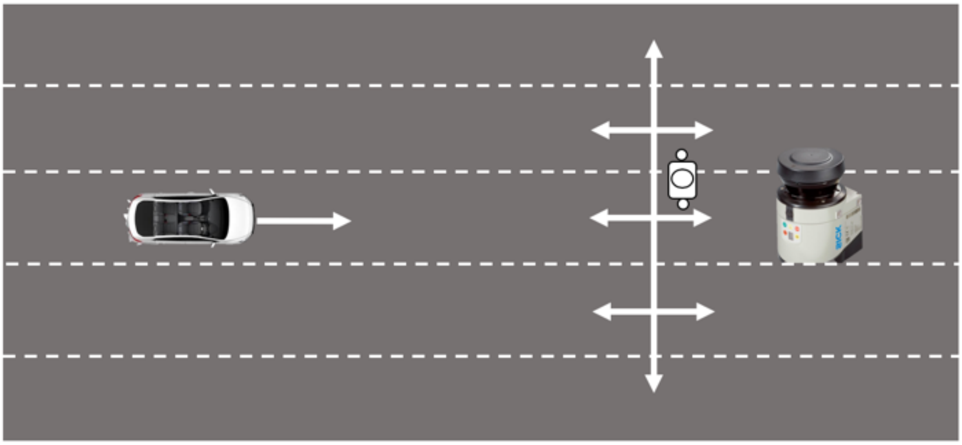

Fig. 9 Experiment for error of estimated states

To evaluate the accuracy of estimated states, we conducted additional experiment to obtain ground truth value. The procedure of experiment is described in Fig 9. A Sick Lidar is utilized for ground truth value. Sick LiDAR is more accurate to measure distance nearby pedestrians than IBEO values moving at long distances.

We measured various movement of pedestrian, and the result is described in Fig. 10 and Table 3. Using fusion algorithm, longitudinal position error reduced drastically.

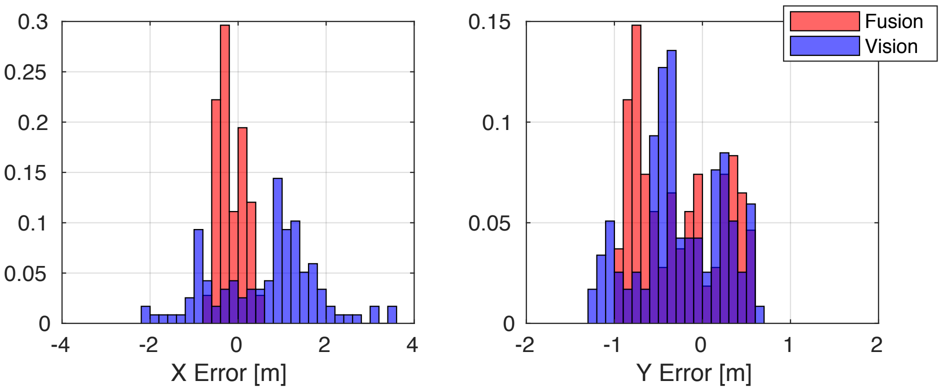

Fig. 10 Histogram of estimated X and Y states error

Table 3 Mean and std of error

	Vision	**Fusion**
X Error mean	0.6196	-0.1565
X Error std	1.1714	0.3027
Y Error mean	-0.2557	-0.2694
Y Error std	0.4843	0.4864

Chapter 7

Conclusion

In this paper, we presented a sensor fusion algorithm using commercial vision sensor, 2D Lidar sensor and Digital HD map. Using MTT algorithm, we robustly tracked a vision sensor data and vision track states set ROI for LiDAR track. MHT algorithm is utilized for LiDAR data tracking algorithm, and two tracks are associated using GNN algorithm. Two track's states are fused by CI algorithm. We validated the algorithm using various driving data. Using an outstanding classification performance of commercial vision sensor and highly accurate point cloud data, we effectively detected pedestrian and tracked the target robustly and accurately regardless movement of a pedestrian. The presented algorithm can be utilized in severely complex urban driving situation to robustly track multi pedestrians. In addition, by installing additional vision sensors, the limits of vision sensor FOV can be overcome.

Bibliography

(1) Wei, P., Cagle, L., Reza, T., Ball, J., & Gafford, J. (2018). LiDAR and camera detection fusion in a real-time industrial multi-sensor collision avoidance system. *Electronics*, *7*(6), 84.

(2) Labayrade, R., Royere, C., Gruyer, D., & Aubert, D. (2005). Cooperative fusion for multi-obstacles detection with use of stereovision and laser scanner. *Autonomous Robots*, *19*(2), 117-140.

(3) Premebida, C., Ludwig, O., & Nunes, U. (2009). LIDAR and vision-based pedestrian detection system. *Journal of Field Robotics*, *26*(9), 696-711.

(4) Cho, H., Seo, Y. W., Kumar, B. V., & Rajkumar, R. R. (2014, May). A multi-sensor fusion system for moving object detection and tracking in urban driving environments. In *2014 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 1836-1843). IEEE.

(5) Chavez-Garcia, R. O., & Aycard, O. (2015). Multiple sensor fusion and classification for moving object detection and tracking. *IEEE Transactions on Intelligent Transportation Systems*, *17*(2), 525-534.

(6) Baig, Q. (2012) Multisensor Data Fusion for Detection and Tracking of Moving Objects from a Dynamic Autonomous Vehicle. PhD thesis, University of

Grenoble1.

(7) Blackman, S. S. (2004). Multiple hypothesis tracking for multiple target tracking. *IEEE Aerospace and Electronic Systems Magazine*, *19*(1), 5-18.

(8) Julier, S. J., & Uhlmann, J. K. (2009). General decentralized data fusion with covariance intersection. *Handbook of multisensor data fusion: theory and practice*, 319-344.

(9) Bar-Shalom, Y. (1990). Multitarget-multisensor tracking: advanced applications. *Norwood, MA, Artech House, 1990, 391 p.*

(10) Blackman, S., & Popoli, R. (1999). Design and analysis of modern tracking systems (Book). *Norwood, MA: Artech House, 1999.*

(11) Thrun, S., Burgard, W., & Fox, D. (2005). *Probabilistic robotics*. MIT press.

(12) Kim, B., Yi, K., Yoo, H. J., Chong, H. J., & Ko, B. (2014). An IMM/EKF approach for enhanced multitarget state estimation for application to integrated risk management system. *IEEE Transactions on Vehicular Technology*, *64*(3), 876-889.

(13) Kazantzis, N. & Kravaris, C. (1999). Time-discretization of nonlinear contorl systems via Taylor methods. *Computers & chemical engineering*, 23, 763-

784.

(14) Bar-Shalom, Y., & Li, X. R. (1995). *Multitarget-multisensor tracking: principles and techniques* (Vol. 19). Storrs, CT: YBs.

(15) Bar-Shalom, Y. (1981). On the track-to-track correlation problem. *IEEE Transactions on Automatic control*, *26*(2), 571-572.

초 록

다중 보행자 추적을 위한 센서 융합 알고리즘 개발

환경 센서를 이용하여 보행자를 인지하고 추적하는 알고리즘은 안전한 도심 자율주행을 위해 가장 중요한 기술 중 하나이다. 본 논문은 상업용 비전 센서, 라이다 센서, 그리고 디지털 지도 정보를 융합해 보행자를 추적하는 새로운 알고리즘을 제시한다. 상업용 비전 센서는 보행자를 효과적으로 탐지하는 반면 라이다 센서는 거리를 정확하게 측정한다. 본 시스템은 상업용 비전 센서를 이용해 보행자를 탐지하며, 라이다 센서를 이용하여 상태 추정 성능을 향상시켰다. 또한 디지털 지도를 이용해 라이다 센서의 관심 영역을 설정하였다. 탐지 결과는 서울대학교 캠퍼스에서 약 4600 프레임 주행 데이터로, 추정의 정확성은 주행 실험을 통해 검증하여 복잡한 도심 주행 상황에서도 본 알고리즘이 유용함을 검증하였다.

주요어: 자율 주행 / 보행자 / 상업용 비전 센서 / 2 차원 라이다 / 디지털 정밀 지도 / 센서 융합 / 다중 대상 추적 / 트랙간 융합

학번: 2017-21074