2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

Mono-Camera based 3D Object Tracking Strategy for Autonomous Vehicles

Akisue Kuramoto, Mohammad A. Aldibaja, Ryo Yanase, Junya Kameyama, Keisuke Yoneda and Naoki Suganuma

Abstract— **This paper proposes an approach to calculate 3D positions of far detected vehicles. Mainly, the distance from the vehicles during autonomous driving must be estimated precisely to strategize a safe path planning. A 3D camera model is created to map the pixel positions to the distance values with respect to the vehicle plane and the distortion parameters. In order to refine the distance accuracy, the Extended Kalman Filter (EKF) framework is designed to track the detected vehicles based on the derivative relationship between the camera and world coordinate systems. The experimental results indicate that the proposed method is capable to successfully track 3D positions with sufficient accuracy compared to LIDAR and Radar based tracking systems in terms of cost and stability.**

I. INTRODUCTION

Autonomous vehicles have received more attention in recent years due to the promising potentials of reducing traffic accidents and transportation fares. In the advance stag of manufacturing, these vehicles are supposed to be deployed on urban roads. Thus, the safe autonomous driving must be guaranteed by achieving high performances of the auto-driver modules. Many researchers have implemented and modified these modules with impressive results such as localization, path planning, motion control and detecting-tracking of surrounding obstacles. The last module is very important to attain safety as it affects the frameworks of the other modules.

The drivable path is generated based on the detected objects in the surrounding environment. Particularly, distance estimation is a critical factor to select the optimal available path and control the curviness of the trajectory. Accordingly, estimating the position of far objects (more than 100m) is very important to properly and smoothly update the path.

Various researches have tried to recognize surrounding obstacles by clustering and labeling the LIDAR point clouds using machine learning techniques [1]. Although the accuracy of estimating position of objects in short to medium distance has been reported to be sufficient for autonomous driving. On the other hand, it is difficult to recognize objects at the range of more than 100m because of the LIDAR scanning range. Millimeter-wave radar (MWR) systems have also been used to the same purpose [2][3]. Such systems allow to confirm the existence of objects at long distances but it is difficult to identify what the objects are. In addition, because of the wave interference, the detection process is not stable and continuous.

Camera image captures distant and close objects with providing 2D shape, texture and color information. The distance to an object can be calculated by modelling the pixel distribution with respect to the camera calibration parameters. In addition, compared with the above mentioned systems, camera based system is preferred as it is commercial and simpler [4].

Relatively, some researchers have tried to classify and estimate object positions in the image domain using hand-designed image features [5][6]. Such methods are limited in number of classified categories and stability. On the other hand, object detection algorithms using deep convolutional neural network (DCNN) has been actively developed. Many DCNN architectures have been proposed to detect and bound the objects in rectangles with high accuracy of classification and location [7][8][9][10][11].

Stereo camera can be used to estimate distance to obstacles. In [12], the objects are described by a set of features and the distance is estimated in range of 200 m, however, it is difficult to classify the detected objects. Another framework is proposed based on edge detection of the objects and the distance accuracy is sufficient in less than 60 m [13]. Mono camera was also tried for the similar purpose in small robots with close obstacles in the range of 50 m [14] [15].

Each of the above study has tried to estimate the object distance in three-dimensional space. Mono-camera provides two-dimensional information which is insufficient to calculate the three-dimensional position of far objects accurately. Even by using multiple cameras, the discretization error of image pixels causes a difference in estimation and reduces the accuracy. However, if the camera position and the surrounding topography are known, it is possible to estimate the three-dimensional position and the discretization error by image pixels can be corrected through the object tracking results. By achieving these two conditions, a mono-camera system can provide a reliable accuracy.

In this paper, we propose a framework for detecting far objects and estimating the corresponding distance to the vehicle using mono-camera and GNSS/IMU system. As this

Akisue Kuramoto is with the Department of Mechanical Systems Engineering, Faculty of Systems Design, Tokyo Metropolitan University, 6-6 Asahigaoka, Hino, Tokyo, 191-0065, Japan (Phone: +81-42-585-8685; e-mail: kuramoto.a.aa@tmu.ac.jp)

Mohammad A. Aldibaja, Ryo Yanase, Keisuke Yoneda and Naoki Suganuma are with the Department of Natural Science, Autonomous Vehicle Research Unit, Institute for Frontier Science Initiative, Kanazawa University, Kakuma-Matchi, Kanazawa, Ishikawa 920-1192, Japan (Phone: +81-76-234-4714; Fax: +81-76-234-4714; e-mail: suganuma@staff.kanazawa-u.ac.jp).

Junya Kameyama is with Sony Semiconductor Solutions Corporation, Atsugi Tec., 4-14-1 Asahi-cho, Atsugi, Kanagawa 243-0014, Japan (e-mail: junya.kameyama@jp.sony.com)

978-1-5386-4451-5/18/$31.00 ©2018 IEEE

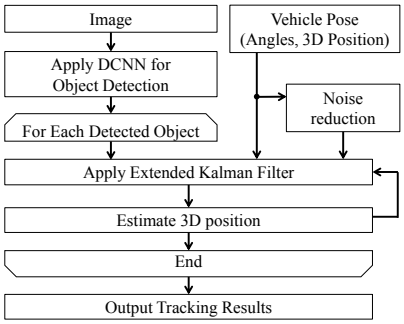


Figure 1. Flowchart of the procedures for detection, distance estimation and tracking of obstacles. Detailed explanation on these steps are provided in the following sections.

study is an initial step in the implementation process, some simple conditions are assumed that the testing environment is flat and the objects are located in the vehicle plane, e.g., parking area. As we generate high definition maps for autonomous vehicles [16], these conditions will be confidently eliminated in future using elevation maps, i.e., the height difference between the vehicle and the detected objects can be easily calculated.

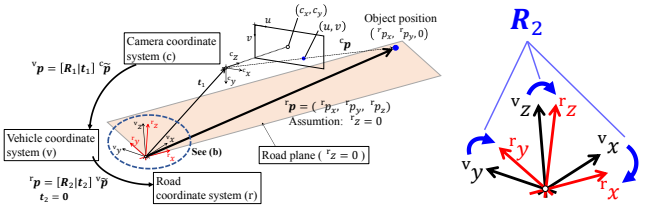
In order to evaluate the system performance, a series of tests has been conducted in the Kanazawa University campus. The experimental results have verified that the proposed framework estimates the object distance reliably and continuously. This fact is emphasized by comparing the measurement results with LIDAR and Radar based perception modules in the range of higher than 100m.

II. Proposal of the New Estimation Method of Distance to Other Traffic Participants

Fig. 1 shows the flowchart of the procedures for 3D position estimation from 2D image domain. The procedures start with capturing image by camera and measuring vehicle pose by GNSS/IMU. A DCNN is used to detect objects in the captured image. The positions of the detected objects are estimated by a 3D camera model. To improve the 3D position accuracy of the detected objects, the Extended Kalman Filter (EKF) is used. Detailed explanation on these steps are provided in the following sections.

A. Data capture

The image is captured by a front camera and inputted to a trained DCNN. The network detects and classifies the objects of vehicles, pedestrians and bicyclists with providing the relevant positions and 2D bounding rectangles in the image domain. As we mainly aim to track the distant objects, estimating distance based only on the bounding rectangles is risky due to the network instability against changing the car type and the background conditions. Therefore, the object position is regarded to be indicated by the 2D bottom center (u, v) of its rectangle in this study. The distance is estimated using a lifting framework from 2D to 3D coordinate system using a lifting framework from 2D to 3D coordinate system with an assumption that the objects are in the same plane of the ego vehicle, e.g., parking areas. Hereinafter, this plane is referred by the term of *Road plane*.



(a) (b)

Figure 2. Relationships between coordinate systems and positions. (a) Overview of the main transformations. (b) Rotation from VCS to RCS.

B. Definition of coordinates systems

In order to estimate the position in the real world, four coordinate systems are used as shown in Fig. 2, i.e., camera coordinates system (CCS), vehicle coordinates system (VCS) and world coordinate system (WCS). Mathematically, the transformation between these systems can be explained by rotation and translation matrices $[\boldsymbol{R}_{th} \quad \boldsymbol{t}_{th}]$ with respect to the origin and the axes directions. According to our assumption, one can observe that the road plane and VCS share the origin coordinates. However, they don't have the same rotation angles and the road coordinates system (RCS) is assumed accordingly. The offsets of the rotation angles between these two systems are estimated using (1) based on the measurements of the GNSS/IMU system. As the measurements may carry some noise, using a digital band-pass filter was technically found to enhance the offset estimation significantly:

$$\Delta\alpha[n] = \frac{\tau(\alpha[n]-\alpha[n-1])+\{2+\tau(\omega_L+\omega_H)\}\Delta\alpha[n-1]-\Delta\alpha[n-2]}{\tau^2\omega_L\omega_H+\tau(\omega_L+\omega_H)+1}, \quad (1)$$

where τ is sampling interval, ω_L and ω_H are lower and higher cutoff frequencies, $\alpha[n]$ is an angle (roll, pitch or yaw) measured on time t_n, and $\Delta\alpha[n]$ is the estimated offset, respectively. Therefore, $\alpha[n] - \Delta\alpha[n]$ is an angle of the estimated road plane. According to $\Delta\alpha[n]$, a rotation matrix from VCS to RCS, $\boldsymbol{R}_2$, is continuously updated.

C. 3D position calculation from 2D image coordinates(3D camera model)

Figure 3 shows the flowchart of estimating the 3D object position in WCS from CCS. Based on the pinhole camera model, the normalized image coordinates (x'', y'') are calculated from (u, v) as follows:

$$\begin{bmatrix} x'' \\ y'' \end{bmatrix} = \begin{bmatrix} (u - c_x)/f_x \\ (v - c_y)/f_y \end{bmatrix}, \quad (2)$$

where c_x, c_y, f_x and f_y are the intrinsic parameters of the camera. In order to obtain the undistorted image coordinates (x', y') from (x'', y''), Levenberg-Marquart Method is applied [17]. Accordingly, a 3D position of an object in CCS, ${}^c\boldsymbol{p} = ({}^cp_x, {}^cp_y, {}^cp_z)$, can be described as follows:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} {}^cp_x / {}^cp_z \\ {}^cp_y / {}^cp_z \end{bmatrix}. \quad (3)$$

As the object is assumed to be in the road plane, (3) can be rewritten to (4).

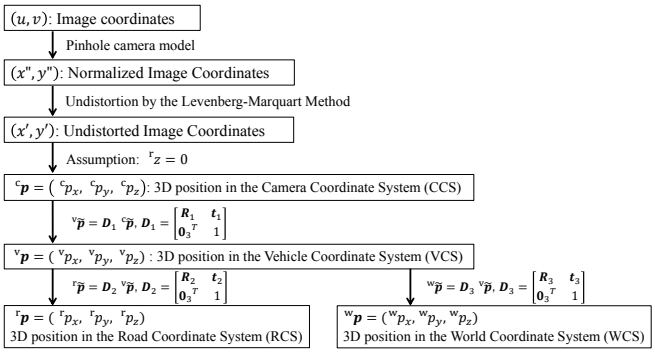


Figure 3. Flowchart of the calculation procedures of 3D position from the 2D image coordinates (u, v)

$$\begin{bmatrix} {}^{c}p_x \\ {}^{c}p_y \\ {}^{c}p_z \end{bmatrix} = \begin{bmatrix} {}^{c}p_z x' \\ {}^{c}p_z y' \\ {}^{c}p_z \end{bmatrix}. \tag{4}$$

Consequently, the 3D position of an object in RCS is expressed as follows:

$$\begin{bmatrix} {}^{r}p_x \\ {}^{r}p_y \\ {}^{r}p_z \end{bmatrix} = \boldsymbol{R}_2[\boldsymbol{R}_1 \quad \boldsymbol{t}_1]\begin{bmatrix} {}^{c}p_z x' \\ {}^{c}p_z y' \\ {}^{c}p_z \\ 1 \end{bmatrix}. \tag{5}$$

As all parameters in the right side of (5) are known except ${}^{c}p_z$, it is calculated using the assumption of ${}^{r}p_z = 0$. Thus, ${}^{c}\boldsymbol{p}$ is obtained by substituting ${}^{c}p_z$ in (4) and, ${}^{v}\boldsymbol{p}$, ${}^{r}\boldsymbol{p}$ and ${}^{w}\boldsymbol{p}$ are simply the 3D affine transformation of ${}^{c}\boldsymbol{p}$. After these calculation, each pixel on image has depth at the moment as shown in Fig. 4.

D. Extended Kalman Filter for Object Tracking

As the distance estimation of far objects is a main objective, the image region closer to the horizon curve has a larger difference between adjacent pixels as shown in Fig. 4. This produces an estimation error of 3D position in WCS. In order to improve the estimation accuracy, Extended Kalman Filter (EKF) is designed based on the camera model outputs and the objects are tracked accordingly. As the framework of EKF is very well known, the designing of the transition and measurement matrices is detailed.

Object in WCS is described by a state vector $\hat{\boldsymbol{X}} = [{}^{w}p_x, {}^{w}p_y, {}^{w}\dot{p}_x, {}^{w}\dot{p}_y, {}^{w}\ddot{p}_x, {}^{w}\ddot{p}_y]^{T}$ which contains the values of position, velocity and acceleration values in x and y directions, respectively. A measurement vector of the object position in the image domain is expressed by $\hat{\boldsymbol{Z}} = [u, v]^{T}$. The process and measurement equations are given as follows:

$$\hat{\boldsymbol{X}}_{t/t-1} = \boldsymbol{f}_t(\hat{\boldsymbol{X}}_{t-1}) + \hat{\boldsymbol{w}}_t, \tag{6}$$

$$\hat{\boldsymbol{Z}}_{t/t-1} = \boldsymbol{h}_t(\hat{\boldsymbol{X}}_{t/t-1}) + \hat{\boldsymbol{v}}_t, \tag{7}$$

where $\hat{\boldsymbol{w}}_t$ and $\hat{\boldsymbol{v}}_t$ represent the process and observation noises. In order to design the transition vector $\boldsymbol{f}_t$, objects are assumed to move with constant accelerations. Thus, $\boldsymbol{f}_t$ is a vector of time-variant functions that correspond to the vector state elements as in (8).

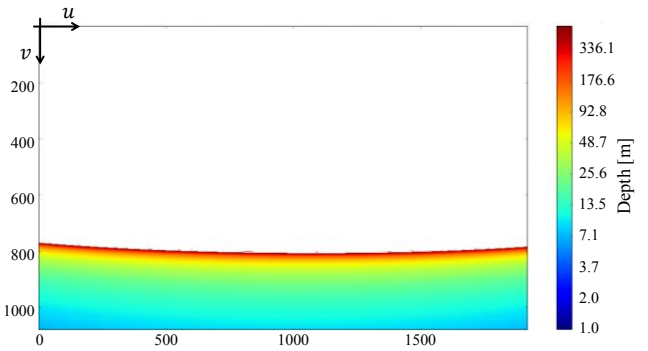


Figure 4. Example of the depth map on an image

$$\boldsymbol{f}_t = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{bmatrix} = \begin{bmatrix} {}^{w}p_x + {}^{w}\dot{p}_x\Delta t + \frac{1}{2}{}^{w}\ddot{p}_x\Delta t^2 \\ {}^{w}p_y + {}^{w}\dot{p}_y\Delta t + \frac{1}{2}{}^{w}\ddot{p}_y\Delta t^2 \\ {}^{w}\dot{p}_x + {}^{w}\ddot{p}_x\Delta t \\ {}^{w}\dot{p}_y + {}^{w}\ddot{p}_y\Delta t \\ {}^{w}\ddot{p}_x \\ {}^{w}\ddot{p}_y \end{bmatrix}, \tag{8}$$

where Δt is a frame interval. In (7), $\boldsymbol{h}_t$ is a set of observation equations in non-linear and time-variant formula. The equations are used to map between the measurement vector $\hat{\boldsymbol{Z}}$ and the estimated $({}^{w}p_x, {}^{w}p_y)$. By considering the camera distortion parameters (k_1, k_2, c_1, c_2) as in (9) and (10), $\boldsymbol{h}_t$ is represented in (11) with respect to the undistorted position (x'', y''):

$$r^2 = x'^2 + y'^2, \tag{9}$$

$$\begin{bmatrix} x" \\ y" \end{bmatrix} = \begin{bmatrix} x'(1 + k_1r^2 + k_2r^4) + 2p_1x'y' + p_2(r^2 + 2x'^2) \\ y'(1 + k_1r^2 + k_2r^4) + p_1(r^2 + 2y'^2) + 2p_2x'y' \end{bmatrix}, \tag{10}$$

$$\boldsymbol{h}_t = \begin{bmatrix} h_1(x'', y'') \\ h_2(x'', y'') \end{bmatrix} = \begin{bmatrix} f_x x'' + c_x \\ f_y y'' + c_y \end{bmatrix}. \tag{11}$$

Accordingly, *Jacobian matrices* $\boldsymbol{F}_t$ and $\boldsymbol{H}_t$ are calculated in (12) and (13).

$$\boldsymbol{F}_t = \frac{\partial \boldsymbol{f}_t}{\partial \hat{\boldsymbol{X}}} = \begin{bmatrix} 1 & 0 & \Delta t & 0 & \Delta t^2/2 & 0 \\ 0 & 1 & 0 & \Delta t & 0 & \Delta t^2/2 \\ 0 & 0 & 1 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 1 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{12}$$

$$\boldsymbol{H}_t = \frac{\partial \boldsymbol{h}_t}{\partial \hat{\boldsymbol{X}}} = \begin{bmatrix} \frac{\partial \boldsymbol{h}_1}{\partial {}^{w}p_x} & \frac{\partial \boldsymbol{h}_1}{\partial {}^{w}p_y} & 0 & 0 & 0 & 0 \\ \frac{\partial \boldsymbol{h}_2}{\partial {}^{w}p_x} & \frac{\partial \boldsymbol{h}_2}{\partial {}^{w}p_y} & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{13}$$

For simplicity, $\boldsymbol{H}_t$ is decomposed into two matrices as follows:

$$\boldsymbol{H}_t = [\boldsymbol{H}'_t \quad \boldsymbol{O}_{2,4}]. \tag{14}$$

In accordance with the law of error propagation, $\boldsymbol{H}'_t$ for numerical calculation is expressed as follows:

$$\boldsymbol{H}'_t = (\boldsymbol{H}_e\boldsymbol{H}_d\boldsymbol{H}_c\boldsymbol{H}_b\boldsymbol{H}_a)^{-1}, \tag{15}$$

where the relationship between the positions and the corresponding derivation are in the series from (16) to (23).

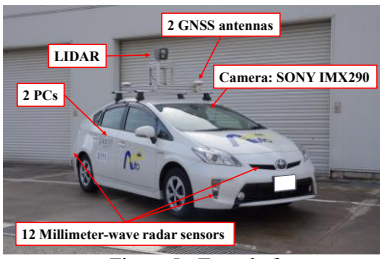


Figure 5. Test platform

First, $\boldsymbol{H}_a$ is the inverse matrix of *Jacobian* of (11) as follows;

$$\boldsymbol{H}_a = \begin{bmatrix} \frac{\partial \boldsymbol{h}_1}{\partial x''} & \frac{\partial \boldsymbol{h}_1}{\partial y''} \\ \frac{\partial \boldsymbol{h}_2}{\partial x''} & \frac{\partial \boldsymbol{h}_2}{\partial y''} \end{bmatrix}^{-1} = \begin{bmatrix} f_x & 0 \\ 0 & f_y \end{bmatrix}^{-1}. \quad (16)$$

$\boldsymbol{H}_b$ represents the inverse matrix of *Jacobian* of transformation from the distorted to undistorted positions of the pixels as in (17);

$$\boldsymbol{H}_b = \begin{bmatrix} \frac{\partial x''}{\partial x'} & \frac{\partial x''}{\partial y'} \\ \frac{\partial y''}{\partial x'} & \frac{\partial y''}{\partial y'} \end{bmatrix}^{-1}, \quad (17)$$

The elements in (17) are calculated by (18) to (20) based on the camera distortion parameters (k_1, k_2, c_1, c_2).

$$\frac{\partial x''}{\partial x'} = (1 + k_1 r^2 + k_2 r^4) + 2x'^2(k_1 + 2k_2 r^2) + 2p_1 y' + 6p_2 x', \quad (18)$$

$$\frac{\partial x''}{\partial y'} = \frac{\partial y''}{\partial x'} = 2x'y'(k_1 + 2k_2 r^2) + 2p_1 x' + 2p_2 y', \quad (19)$$

$$\frac{\partial y''}{\partial y'} = (1 + k_1 r^2 + k_2 r^4) + 2y'^2(k_1 + 2k_2 r^2) + 6p_1 y' + 2p_2 x'. \quad (20)$$

$\boldsymbol{H}_c$ represents the *Jacobian* of ${}^{c}\boldsymbol{p}$ by the undistorted image coordinates (x', y') as in (21);

$$\boldsymbol{H}_c = \begin{bmatrix} \frac{\partial\, {}^{c}p_x}{\partial x'} & \frac{\partial\, {}^{c}p_x}{\partial y'} \\ \frac{\partial\, {}^{c}p_y}{\partial x'} & \frac{\partial\, {}^{c}p_y}{\partial y'} \\ \frac{\partial\, {}^{c}p_z}{\partial x'} & \frac{\partial\, {}^{c}p_z}{\partial y'} \end{bmatrix} = {}^{c}p_z \begin{bmatrix} (1 + \frac{q_{31}}{q_z} x') & \frac{q_{32}}{q_z} x' \\ \frac{q_{31}}{q_z} y' & (1 + \frac{q_{32}}{q_z} y') \\ \frac{q_{31}}{q_z} & \frac{q_{32}}{q_z} \end{bmatrix}, \quad (21)$$

where q_{ij} is an element of the $\boldsymbol{R}_2\boldsymbol{R}_1$, and q_z is the z coordinate of the $\boldsymbol{R}_2\boldsymbol{t}_1$. Note that ${}^{c}p_z$ can be obtained by the proposed 3D camera model. $\boldsymbol{H}_d$ represents the relationship between vehicle and camera coordinate systems.

$$\boldsymbol{H}_d = \begin{bmatrix} \frac{\partial\, {}^{v}p_x}{\partial\, {}^{c}p_x} & \frac{\partial\, {}^{v}p_x}{\partial\, {}^{c}p_y} & \frac{\partial\, {}^{v}p_x}{\partial\, {}^{c}p_z} \\ \frac{\partial\, {}^{v}p_y}{\partial\, {}^{c}p_x} & \frac{\partial\, {}^{v}p_y}{\partial\, {}^{c}p_y} & \frac{\partial\, {}^{v}p_y}{\partial\, {}^{c}p_z} \\ \frac{\partial\, {}^{v}p_z}{\partial\, {}^{c}p_x} & \frac{\partial\, {}^{v}p_z}{\partial\, {}^{c}p_y} & \frac{\partial\, {}^{v}p_z}{\partial\, {}^{c}p_z} \end{bmatrix} = \boldsymbol{R}_1. \quad (22)$$

Finally, $\boldsymbol{H}_e$ encodes the derivative conversion from vehicle to world coordinate systems as in (22);

$$\boldsymbol{H}_e = \begin{bmatrix} \frac{\partial\, {}^{w}p_x}{\partial\, {}^{v}p_x} & \frac{\partial\, {}^{w}p_x}{\partial\, {}^{v}p_y} & \frac{\partial\, {}^{w}p_x}{\partial\, {}^{v}p_z} \\ \frac{\partial\, {}^{w}p_y}{\partial\, {}^{v}p_x} & \frac{\partial\, {}^{w}p_y}{\partial\, {}^{v}p_y} & \frac{\partial\, {}^{w}p_y}{\partial\, {}^{v}p_z} \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \end{bmatrix}, \quad (23)$$

where r_{ij} is an element of the $\boldsymbol{R}_3$, and t_z is the z coordinate of the $\boldsymbol{t}_1$. According to $\boldsymbol{H}'_t$, $\hat{\boldsymbol{X}}_{t-1/t-1}$ and $\hat{\boldsymbol{Z}}_{t/t-1}$, the $\hat{\boldsymbol{X}}_{t/t-1}$ and its error covariance matrix at the each time t are estimated.

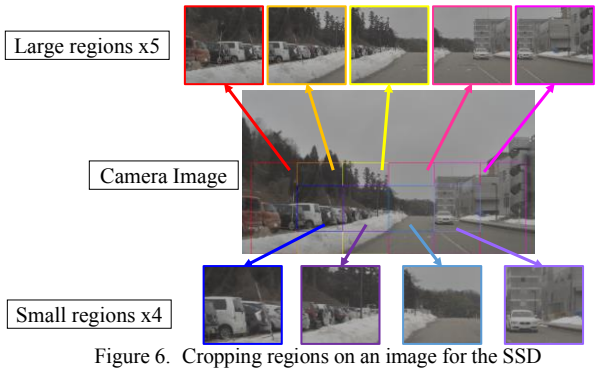


Figure 6. Cropping regions on an image for the SSD

III. Platform and Settings of Experiment

A. Experimental Platform

Fig. 5 shows the experimental vehicle equipped with many sensors and devices. A camera system using SONY IMX290 image sensor is attached to the windshield. The camera output is a Full HD (1,920 x 1,080) image with 7.5 fps and the frame interval is $\Delta t = 0.133$ s. Two GNSS antennas are attached to the vehicle roof and Applonix POS-LV 220 coupled GNSS/IMU is installed to receive GPS data and measure the velocity, acceleration and rotation angles. Two PCs with Intel-Core™ i7-6700 CPU working at 3.40 GHz with 8 GB of RAM are deployed in the trunk. One of the PCs is equipped with a NVIDIA GTX-1080Ti GPU with water cooling system and specialized for applying DCNN for object detection; the VGG-16-trained Single Shot MultiBox Detector (SSD) [7][10] is used as well. The operating system is Ubuntu 14.04 and the object detection function was coded in Python 2.7.6 with using Keras 2.0.4 whose backend was Tensorflow v1.1.0. The data from multiple sensors and the object detection results are received by the other PC and the calculation of the object tracking is calculated accordingly. Velodyne HDL-64E S2 laser range finder LIDAR with 64 separate beams is attached to the vehicle roof. Twelve millimeter-wave radars (MWR) are distributed on the vehicle body to scan distant areas in range of 180 m. In the next chapter, the results of object tracking by these three systems are used to evaluate the proposed system.

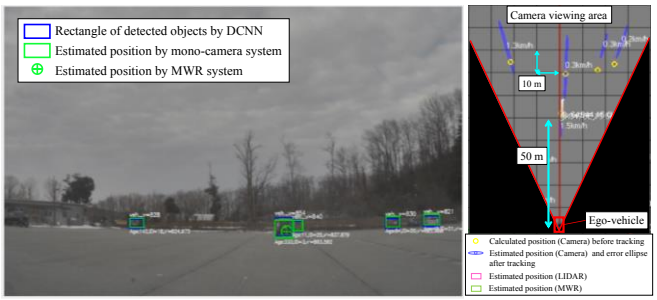
B. Settings

The camera image is divided into nine regions with two different sizes as shown in Fig. 6. The region distribution was determined empirically to represent the road in the driving direction, and the number of regions was determined based on processing the nine resized images by SSD within the frame interval. All regions are resized into 300 x 300 to be compatible with the SSD input size. Accordingly, the candidates of surrounding rectangles are managed to best bounding the detected objects [10].

GNSS/IMU measurements are obtained with 100 Hz and the offset calculation between WCS and RCS was performed in the sampling interval $\tau = 0.01$ s. As τ differs from Δt, the vehicle pose and RCS were obtained in the closest time to the time image captured were applied to the calculation on each time. For the road plane estimation, ω_L and ω_H were respectively set to 0.5 Hz and 1.0 Hz to reduce noises that caused by measurement error and road bump.

TABLE I. TEST SCENARIO

Synbol	*Target objects*	*Test vehicle state*	*Location*
A	Stopping vehicles	Run	Parking area
B	Oncoming vehicle	Run	Straight road
C	Preceding vehilce	Stopped	Straight road



(a) (b)

Figure 7. Test scene in Scenario-A. (a) captured image and detected objects. (b) position map of detected objects.

The proposed system has been tested and evaluated using data that captured at almost-flat areas: a parking lots and a straight road in Kanazawa University campus. Three test scenarios were conducted as described in Table1. In each scenario, the distance to the target object was estimated/compared by the proposed system, MWR and LIDAR. The result videos on scenarios B and C can be checked via the following link: https://drive.google.com/open?id=1NA92Qkhe5cfcRjh4K4QEhmcitTb7WNrS.

IV. RESULTS AND DISCUSSION

In this section, we consider the MWR measurements as the *ground truth* because the scanning capability is up to 180m whereas LIDAR enables the maximum scanning range at 90m. These two ranges become more confident when the tracking algorithm is applied in the dynamic environments. The sensing (and tracking) frequencies of MWR and LIDAR were 30 Hz and 10 Hz. As they are different from the camera frame rate, it is difficult to compare tracking results directly. Therefore, the estimated positions by MWR and LIDAR at the time of image were calculated by interpolating tracking results by each sensor at the two closest times.

In Scenario-A, the reliability of the road plane assumption was confirmed as illustrated in Fig. 7. Five parked vehicles exist in front of the ego vehicle as shown in Fig. 7a. Because of the static environmental conditions and due to some noise in the data association techniques, MWR and LIDAR detect only the closest car whereas the enhanced SSD detects the five cars with providing the relevant bounding rectangles. Accordingly, the corresponding positions are estimated by the proposed system. The positions before and after applying EKF as highlighted by the centers of yellow circles and of blue ellipse in Fig. 7b. Comparing with MWR and LIDAR distance results on the closest car, the proposed system estimates the almost the same distance at 50m (with less than 1 m difference) as shown in Fig. 7b. This result emphasizes the reliability to use the road plane assumption for estimating a 3D position of an object. In addition, compared to LIDAR and MVR systems, the other cars are only detected by SSD and the corresponding

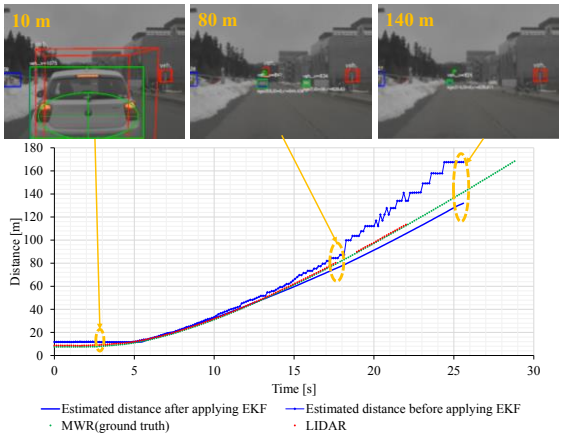


Figure 8. Tracking results and captured image in Scenario-B

distances are estimated by the proposed system. This implies that the proposed system has recognition capability of objects in wider distance range than other sensors.

In Scenario-B, the ego vehicle was static and tracking a proceeding vehicle is conducted. The LIDAR, MWR and proposed systems tracked the vehicle up to 113 m, 168 m and 132 m, respectively and as shown in Fig. 8. On can observe that the LIDAR (red) and MWR (green) estimations are almost identical. The magenta profile represents the estimated distance using the 3D camera model only. The model provides impressive results up to 80m. In addition, SSD detected the vehicle continually for a longer range than LIDAR. After this distance the error occurred because the proceeding vehicle was gradually leaving the road plane of the stopped ego vehicle. However, by applying the tracking system, the error has been significantly reduced to less than 10 m at the distance of 140m (about 7% error).

In order to evaluate the proposed system in a dynamic environment with real traffic conditions, Scenario-C was conducted. The ego vehicle moves and another vehicle is approaching that the distance becomes zero at the intersection point. In this case, the road plane parameters are changed according to the ego vehicle location as illustrated in Fig. 9. The corresponding effects on the distance estimation using only the 3D camera model is highlighted in Fig. 10, i.e., dot profiles: blue/bright w/o considering the change of parameters). By applying the tracking system, these two profiles are refined and become more stable as shown by the corresponding solid profiles. Estimated distance with the road plane assumption is apparently closer to the ground truth than without the road plane assumption. The MWR system tracked the oncoming vehicle initially and then lost it until to 90 m. In addition, LIDAR system started to track the vehicle at 56 m. This is because of the low capability of detecting the objects at far distance. This situation is exactly in contrast of distance and then tracked for long time. According to the above discussion, the proposed system provides a very stable performance of detection and tracking. In addition, comparing to MWR (ground truth) in the active tracking area, the Scenario-B where the vehicle was easily detected at near proposed system provides a very reasonable accuracy with maximum estimation error of 8 m as shown in Fig. 10. These results show that the proposed system can recognize the type and approximate position of far objects simultaneously, and can increases the position accuracy as the objects approach.

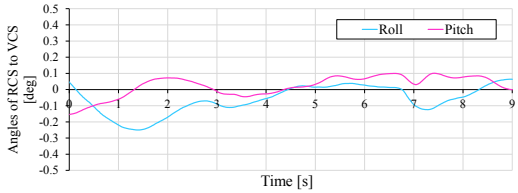


Figure 9. Angles of RCS from VCS during an oncoming vehicle is detected on images in Scenario-C.

V. CONCLUSION

Distance estimation of the surrounding vehicles is still a very challenging demand for conducting safe autonomous driving especially for camera based tracking system. In this paper, a framework of using 3D camera model and EKF are designed and evaluated to calculate the vehicle positions in the real world. Under the assumption that surrounding objects are in the plane parallel to the road surface, the rational offsets between these two domains are continuously updated. This assumption enables to estimate the distance accurately. The output of the camera model is interlay utilized to calculate the measurement matrix of the Extended Kalman Filter (EKF). The matrix is designed to map between the position measurement on the detected vehicles in the image domain and the corresponding vector state in the real world. Accordingly, the vehicle is tracked and the 3D position is estimated with the relevant covariance matrix of the error. The experimental results emphasize the reliability of the proposed method with respect to the considered assumption. In addition, a very reasonable distance accuracy of the far vehicle has been obtained. The tracking stability has been evaluated and compared to LIDAR and Radar systems. The proposed method relatively performed a robust tracking profile in a dynamic and flat environment with different initial tracking conditions.

Based on the promising results, the proposed framework can be integrated to estimate the distance of vehicles that aren't in the same plane of the ego-vehicle. This can be achieved by using elevation maps which provide the height information. Accordingly, the height between the ego-vehicle plane and the real world/other-vehicle can be obtained and the corresponding positions are estimated precisely.

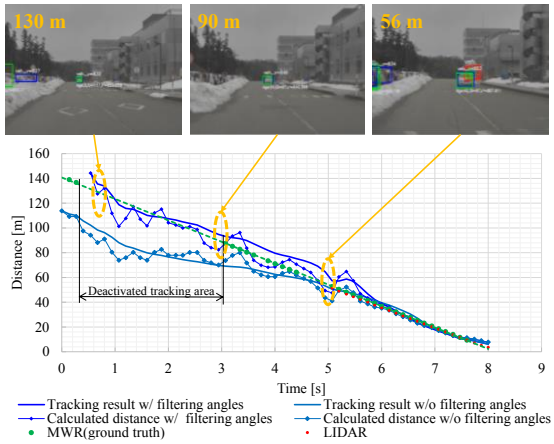


Figure 10. Tracking results and captured image in Scenario-C

REFERENCES

[1] Himmelsbach, M., Luettel, T. and Wuensche, H. J. "Real-time object classification in 3D point clouds using point feature histograms." In *IEEE/RSJ International Conference on Intelligent Robots and Systems,* IEEE, 2009. pp. 994-1000.

[2] Sugimoto, S., Tateda, H., Takahashi, H. and Okutomi, M. "Obstacle detection using millimeter-wave radar and its visualization on image sequence." In *Proceedings of the 17th International Conference on Pattern Recognition (ICPR)*, IEEE, 2004. pp.342-345.

[3] Kaliyaperumal, K., Lakshmanan, S. and Kluge, K. "An algorithm for detecting roads and obstacles in radar images." *IEEE Transactions on Vehicular Technology*, 50(1), 2001. pp. 170-182.

[4] J. Ziegler, H. Lategahn, M. Schreiber, C. G. Keller, C. Knoppel, J. Hipp, M. Haueis and C. Stiller. "Video based localization for BERTHA", In *2014 IEEE Intelligent Vehicles Symposium Proceedings*,IRRR, 2014. pp. 1231-1238.

[5] Broggi, A., Caraffi, C., Fedriga, R. I. and Grisleri, P. "Obstacle detection with stereo vision for off-road vehicle navigation." In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops*, IEEE, 2005. pp. 65.

[6] Wang, B., Florez, S. A. R. and Vincent Fremont. "Multiple Obstacle Detection and Tracking using Stereo Vision: Application and Analysis." In *13th International Conference on Control, Automation, Robotics & Vision (ICARCV),* IEEE, 2014, pp.1074-1079.

[7] Simonyan, K and Zisserman, A. "Very Deep Convolutional Networks for Large-Scale Image Recognition." *arXiv preprint arXiv:1409.1556*, 2014.

[8] Ren, S., He, K., Girshick, R. and Sun, J. "Faster r-cnn: Towards real-time object detection with region proposal networks." In *Advances in neural information processing systems*, pp. 91-99, 2015.

[9] Redmon, J. and Farhadi, A. "YOLO9000: Better, Faster, Stronger." In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.6517-6525, 2017.

[10] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C., "SSD: Single Shot Multibox Detector." In *European Conference on Computer Vision – ECCV 2016. Lecture Notes in Computer Science*, Vol 9905. pp. 21-37, Springer, Cham 2016.

[11] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., E, Dumitru., Vanhoucke, V. and Rabinovich, A. "Going Deeper with Convolutions." *arXiv preprint arXiv: 1409.4842*, 2017.

[12] Pinggera, P., Franke, U. and Mester, R. "High-performance long range obstacle detection using stereo vision." In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),* IEEE, 2015. pp. 1308-1313.

[13] Nedevschi, Sergiu, et al. "High accuracy stereo vision system for far distance obstacle detection." In *Intelligent Vehicles Symposium, 2004 IEEE.* IEEE, 2004. pp. 292-297

[14] Lee, T. J., Yi, D. H. and Cho, D. I. "A Monocular Vision Sensor-Based Obstacle Detection Algorithm for Autonomous Robots." *Sensors*, 2016, 16, 311.

[15] Lin, C. C. and Wang, M. S., "A vision based top-view transformation model for a vehicle parking assistant." *Sensors*, 2012, 12, pp. 4431-4446.

[16] Mohammad, A., Suganuma, N. and Yoneda, K. "LIDAR-data accumulation strategy to generate high definition maps for autonomous vehicles." in *Multisensor Fusion and Integration for Intelligent Systems (MFI), 2017 IEEE International Conference on*. IEEE, Nov. 2017, pp. 422-428.

[17] Levenberg, Kenneth (1944). "A Method for the Solution of Certain Non-Linear Problems in Least Squares". *Quarterly of Applied Mathematics*, 1944, 2(2), pp. 164–168.

Again by the assumption that an object is on the road plane (${}^{r}p_z = 0$), the ${}^{w}\boldsymbol{p}$, ${}^{r}\boldsymbol{p}$, ${}^{v}\boldsymbol{p}$, ${}^{c}\boldsymbol{p}$ and (x', y') is obtained numerically.

In the scene (B), the driver of preceding vehicle was asked to go at the speed of about 20 km/h. In the scene (C), drivers of both running cars were asked to go at the speed of about 30 km/h respectively. The results and images presented in this paper were obtained in 25th December 2017 and 22nd January 2018. The weather was cloudy and the road surface was fine and there was snow beside the road.