INTECH
open science | open minds

OPEN ACCESS
ARTICLE

International Journal of Advanced Robotic Systems

Stereo-Based Tracking-by-Multiple Hypotheses Framework for Multiple Vehicle Detection and Tracking

Regular Paper

Young-Chul Lim[1,2], Jonghwan Kim[1], Chung-Hee Lee[1] and Minho Lee[2,*]

1 Division of Advanced Industrial Science & Technology, Daegu Gyeongbuk Institute of Science & Technology, Daegu, Republic of Korea
2 School of Electric & Electrical Engineering, Kyungpook National University, Daegu, Republic of Korea
* Corresponding author E-mail: mholee@knu.ac.kr

Received 23 Aug 2012; Accepted 23 May 2013

DOI: 10.5772/56688

Abstract In this paper, we present a tracking-by-multiple hypotheses framework to detect and track multiple vehicles accurately and precisely. The tracking-by-multiple hypotheses framework consists of obstacle detection, vehicle recognition, visual tracking, global position tracking, data association and particle filtering. The multiple hypotheses are from obstacle detection, vehicle recognition and visual tracking. The obstacle detection detects all the obstacles on the road. The vehicle recognition classifies the detected obstacles as vehicles or non-vehicles. 3D feature-based visual tracking estimates the current target state using the previous target state. The multiple hypotheses should be linked to corresponding tracks to update the target state. The hierarchical data association method assigns multiple tracks to the correct hypotheses with multiple similarity functions. In the particle filter framework, the target state is updated using the Gaussian motion model and the observation model with associated multiple hypotheses. The experimental results demonstrate that the proposed method enhances the accuracy and precision of the region of interest.

Keywords Stereo Vision, Multiple Object Tracking, Bayesian Filter, Multiple Hypotheses, Data Association

1. Introduction

In order for vehicles to navigate automatically, it is very important to perceive the external environment accurately and reliably with object detection and tracking. These factors require various expensive sensors such as radar, lidar, cameras, and GPS to perceive the external environment accurately. Actually, Team Tartan Racing's vehicle "Boss", which was Carnegie Mellon University's winning entry in the 2007 DARPA Urban Challenge, is equipped with 13 different perception sensors [1]. Various vehicle detection methods [2] for intelligent vehicle fields have been introduced in recent decades, and many algorithms and systems have been reported and demonstrated to enhance the reliability and robustness of these systems.

Range sensors, such as lidar [3] and radar [4], have been used as standard approaches for robust object detection

and localization systems. However, these sensors give only point information for the detected target and it is very difficult to recognize the class of the detected object. Camera-based perception methods have been proposed to detect and recognize moving objects while localizing the object's position with prior perspective information [5, 6]. Currently, many researchers have been working toward stereo vision-based approaches in order to provide systems with more reliable detection and localization performance [7-11].

No state-of-the-art detection and recognition algorithms [2, 12] can detect and recognize all the objects on the road without false alarms. Recently, several multiple object tracking methods have integrated object detectors and visual trackers to provide reliable object detection and localization output [13-15]. In [13], an integrated system, a WaldBoost detector [16] and tracking-learning-detection (TLD) [17], is proposed to detect and track vehicles in real-time using a single camera. However, the method has a delay in confirming a target object, because the detector runs every three frames and three consecutive right detections are required. The tracking-by-detection framework utilizes the output of an object detector as an observation model of a Bayesian filter [15]. The framework reduces the number of false detections while enhancing the detection probability. The tracking-by-detection framework reduces false detections during track initialization due to the sparse occurrence of false alarms. The framework can also increase the detection probability while estimating the target object state with a visual tracker, such as a Kanade-Lucas-Tomasi (KLT) [14] and a particle filter [15] when the detector misses an object in the current frame. A tracking method using a particle filter was proposed to re-initialize the tracking algorithm automatically whenever the performance severely deteriorates [18]. In [15], the framework consists of an object-specific detector, a visual tracker and data association for tracking multiple objects. The region of interest (ROI) is updated by the particle filter with a motion model and observation model. A constant velocity model is used for the motion model and associated detection output and an online classifier are used for an output observation model. However, the updated ROI is mainly dependent on the output of the associated detection because the motion model in an image plane is inaccurate due to the nonlinearity of the target's movement. Only a very small number of works [7, 14] have introduced a stereo-based tracking-by-detection framework for detecting and tracking multiple vehicles.

Stereo-based multiple object tracking methods have an advantage in that they can localize objects not only in the 2D image plane but also in 3D global coordinates. A method using an occupancy grid and interacting multiple model (IMM) filter [11], and methods [8-10] that combine depth and motion information have been proposed to detect and track multiple vehicles or pedestrians using 3D information. The state of a target vehicle, including its position, orientation, velocity, acceleration and yaw rate, is estimated while tracking a 3-D point cloud in global coordinates [9]. The method can automatically detect the target object by using a fusion method with vision and radar. In [11], the researchers reconstructed 3D points using a depth image and mapped them onto an occupancy grid using an inverse sensor model. Clustered and segmented objects are associated with tracks and served as an input of the IMM filter. The method extracts obstacles on a road using an occupancy grid and it does not classify specific target objects such as pedestrians or vehicles.

In the field of intelligent vehicles, most stereo-based multiple object tracking methods have been concerned with object detection and localization problems in 3D global coordinates [8-11]. There has been a lack of efforts to increase the accuracy and precision of the ROI in the image plane. In order to enhance the precision as well as the accuracy of the ROI, we propose a tracking-by-multiple hypotheses framework based on the Bayesian probability model. The proposed method uses a hierarchical data association method, 3D feature-based visual tracking and a particle filter using associated multiple hypotheses. The particle filter updates a target state with not only vehicle recognition outputs, but also obstacle detection and visual tracking outputs.

This paper is structured as follows. Our stereo-vision system and tracking-by-multiple hypotheses framework are introduced in Section 2. In Section 3, the proposed multiple vehicle tracking approach using tracking-by-multiple hypotheses is described. This framework consists of a global position tracking, 3D feature-based tracking, hierarchical data association and a particle filter. A qualitative evaluation metric is detailed, and experimental results and analysis are presented in Section 4. Finally, Section 5 provides the conclusion and insight for future works.

2. System overview

2.1 Stereo vision system for intelligent vehicles

Our stereo vision system consists of stereo matching, obstacle detection, vehicle recognition and multiple object tracking modules, as shown in Fig. 1. The stereo matching module, based on the belief propagation algorithm [19], is implemented on the embedded platform with FPGA for real-time processing. The stereo matching module offers two grey images (left and right images) and a depth image to the software platform with VGA @15fps. The dense depth image has 128 disparity levels.

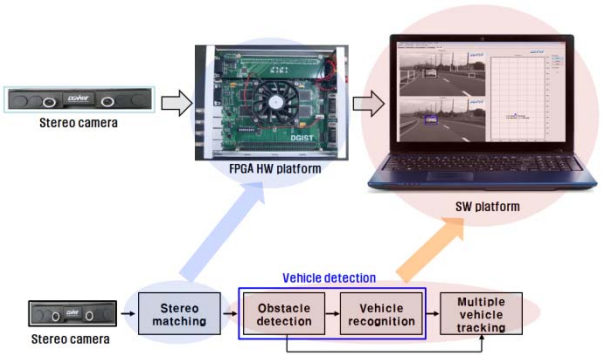


Figure 1. Architecture of stereo vision system

The obstacle detection module extracts the road information using the v-disparity method [20] and then detects all the obstacles on the road using a disparity histogram [21]. The vehicle recognition module classifies the obstacles as vehicle or non-vehicle using the cascaded AdaBoost algorithm [22]. Searching regions are restricted according to the region determined from the obstacle detection module [23]. This approach not only removes false positive alarms, but also reduces the computation time for vehicle detection. The number of false positive alarms can be drastically reduced in the recognition module. On the other hand, vehicle detection probability is slightly decreased by the errors of obstacle detection and vehicle recognition. The multiple vehicle tracking module updates the state (global position and velocity, ROI position and size) of a vehicle and minimizes the number of false alarms caused by the imperfect obstacle detection and vehicle recognition algorithms. One of the advantages of the stereo vision system is that the global position and motion of the target object can be estimated accurately and reliably; also, this system is very helpful for distinguishing between the target object and other objects [14].

2.2 Tracking-by-multiple hypotheses framework

The tracking-by-multiple hypotheses framework consists of obstacle detection, vehicle recognition, global position tracking, visual tracking, data association and a particle filter, as shown in Fig. 2.

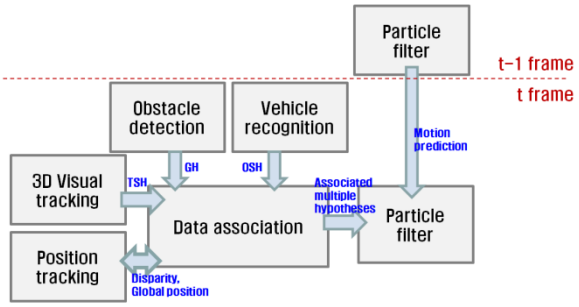


Figure 2. Block diagram of tracking-by-multiple hypotheses

In global position tracking, the accurate sub-pixel disparity of the object can be calculated using the stripe-based accurate disparity (S-BAD) estimation method; the global 3D position and velocity of an object can be updated using the inverse perspective mapping-based extended Kalman filter (IPM-based EKF) [24].

Feature-based visual tracking enables the ROI of the current target object to be estimated from the previous ROI. In feature-based tracking, one of the difficult problems is to find corresponding feature pairs in the current image. Another important point is removing the outlier features corresponding to other objects or to the background. The Kanade-Lucas-Tomosi (KLT) [25] feature tracker has been widely used to deal with real-time tracking problems due to its fast computation and generality [26-27]. However, the KLT is vulnerable to severe illumination change or abrupt object movement. Also, it easily fails to track a target when there are many outlier features in a cluttered environment. Our 3D feature-based tracking method is proposed so as to overcome these problems.

All the existing tracks are connected to correct observations in order to update the target states (global position and velocity, ROI position and size) in a multiple object tracking problem. A hierarchical data association approach deals with the track-to-multiple hypotheses assignment problem. The hierarchical data association utilizes the sub-pixel disparity and global position of the global position tracking module [24], outputs of the visual tracking module, outputs of obstacle detection module [21], and outputs of the vehicle recognition module [23] to assign multiple hypotheses to multiple tracks. In [14], the association cost is calculated by considering the similarity of the sub-pixel disparity and the longitudinal and lateral distance. In this work, we improve the robustness of data association by adding the criteria of the local distance and appearance similarity.

The ROI update module utilizes three types of hypotheses from the following respective modules: obstacle detection, vehicle recognition, and visual tracking modules. They are designated as general hypothesis (GH), object-specific hypothesis (OSH), and target-specific hypothesis (TSH), respectively. GH gives a very high detection probability, but provides poor ROI precision and a high false positive alarm rate. OSH has the advantage of removing many false positive alarms from GH and improving the ROI precision. The number of false negative alarms increases slightly and GH often provides noisy and unstable ROI outputs. The ROI of TSH is very dependent on the ROI states of the tracking object and the track drifting problem often occurs when tracking a target for a long time without GH or TSH. The particle filter using Bayesian probability updates the current ROI with the associated multiple hypotheses in order to enhance the ROI precision and accuracy.

3. Multiple vehicle tracking using tracking-by-multiple hypotheses framework

3.1 Global position tracking with IPM-based EKF

Global position tracking estimates the position and velocity of a target object on the road using a stereo vision system. The accuracy of longitudinal distance mainly depends on the accuracy of disparity; accurate disparity estimation is very important in estimating the distance accurately and precisely. In [24], we proposed the S-BAD estimation method to accurately and reliably estimate sub-pixel disparity. The experimental results show that the proposed method can estimate the sub-pixel disparity with about 0.1 pixel error as well as a distance of less than 50 m with approximately 2% error.

The IPM-based EKF method reduces the error covariance of the position and velocity of the target. In the prediction step of EKF, a system equation, a state transition matrix ($F_{k/k-1}$) with a constant velocity model, and a state vector ($x_{i,k}$) of i^{th} track with position and velocity are defined by

$$
\begin{aligned}
&x_{i,k} = F_{k/k-1}x_{i,k-1} + w_k,\\
&\begin{bmatrix} X_{i,k} \\ \dot{X}_{i,k} \\ Z_{i,k} \\ \dot{Z}_{i,k} \end{bmatrix} = \begin{bmatrix} 1 & dt & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & dt \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_{i,k-1} \\ \dot{X}_{i,k-1} \\ Z_{i,k-1} \\ \dot{Z}_{i,k-1} \end{bmatrix} + w_k,\\
&w_k \sim N(0, Q_k),\\
&\hat{x}^-_{i,k} = F_{k/k-1}\hat{x}^+_{i,k-1},\\
&\hat{P}^-_{i,k} = F_{k/k-1}\hat{P}^+_{i,k-1}F^T_{k/k-1} + Q_{k-1},\\
&Q_{k-1} = \begin{bmatrix} \frac{\eta_x{}^2 dt^4}{4} & \frac{\eta_x{}^2 dt^3}{2} & 0 & 0 \\ \frac{\eta_x{}^2 dt^3}{2} & dt^2 & 0 & 0 \\ 0 & 0 & \frac{\eta_z{}^2 dt^4}{4} & \frac{\eta_z{}^2 dt^3}{2} \\ 0 & 0 & \frac{\eta_z{}^2 dt^3}{2} & dt^2 \end{bmatrix},
\end{aligned}
\tag{1}
$$

where w_k and Q_k are the process noise and the process noise covariance, respectively. η_x and η_z are standard deviations of acceleration noise in the lateral and longitudinal direction, and they are set to 1 and 5 in our experiment, respectively. *dt* is the update time, $\hat{P}^-_{i,k}$ and $\hat{P}^+_{i,k}$ are, respectively, *a priori* and *posterior* error covariance matrixes of the i^{th} track, $\hat{x}^-_{i,k}$ and $\hat{x}^+_{i,k}$ denote *a priori* and *posterior* state vectors of the i^{th} track, respectively. The state vector does not consider the *Y* direction because only obstacles on the road are considered in this work. In the observation step, the observation equation becomes complicated due to the many variables in the IPM model. The equation is simplified by considering the primary variables such as the sub-pixel disparity and the horizontal position of the image.

$$
\begin{aligned}
&z_{j,k} = h(x_{i,k}) + v_k,\\
&v_k \sim N(0, R_k),\\
&R_k = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_d^2 \end{bmatrix},\\
&z_{j,k} = [x_{dl}\ d_{acc}]^T,\\
&z^-_{j,k} = h(\hat{x}^-_{i,k}) = [h_1\ h_2]^T,\\
&h_2 = (b\alpha\cos\theta - y_d\sin\theta) / \hat{x}^-_{i,k}(3),\\
&h_1 = [2\alpha\hat{x}^-_{i,k}(1) / \{(Y_g + h)\sin\theta + \hat{x}^-_{i,k}(3)\cos\theta\} + h_2] / 2,\\
&x_{dl} = x_{pl} - x_o, x_{dr} = x_{dl} - d, y_d = y_{pl} - y_o,
\end{aligned}
\tag{2}
$$

where $h(x)$ is a nonlinear measurement function and R_k is the measurement noise covariance. σ_x and σ_d denote standard deviations of measurement noise and they are set to 1 and 0.5, respectively. *d*, *b*, and *h* are the disparity, baseline and the height of the camera, respectively. θ denotes the angle between the *Z* direction and the optical axis of the cameras, and α is the focal length expressed in units of pixel length. Y_g is the *Y* position of the object in global coordinates, and x_{pl} and y_{pl} indicate the object position in the left image coordinates. x_o and y_o denote the optical centre of the camera. $\hat{x}^-_{i,k}(n)$ denotes the n^{th} element of the state vector. x_{dl} and d_{acc} are the horizontal position on the left image and the sub-pixel disparity, respectively. The observation matrix (*H*) is represented by a Taylor series of a nonlinear function; the higher-order terms in the expansion can be ignored. The first-order approximation coefficients are calculated using a Jacobian matrix [24].

$$
\begin{aligned}
&H_{j,k} = \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{21} & h_{22} & h_{23} & h_{24} \end{bmatrix} = \begin{bmatrix} \frac{\partial h_1}{\partial x_{i,k}(1)} & \frac{\partial h_1}{\partial x_{i,k}(2)} & \frac{\partial h_1}{\partial x_{i,k}(3)} & \frac{\partial h_1}{\partial x_{i,k}(4)} \\ \frac{\partial h_2}{\partial x_{i,k}(1)} & \frac{\partial h_2}{\partial x_{i,k}(2)} & \frac{\partial h_2}{\partial x_{i,k}(3)} & \frac{\partial h_2}{\partial x_{i,k}(4)} \end{bmatrix},\\
&h_{11} = \alpha / \{(Y_g + h)\sin\theta + x^-_{i,k}(3)\cos\theta\},\\
&h_{13} = (-\cos\theta) / \{(Y_g + h)\sin\theta + x^-_{i,k}(3)\cos\theta\}^2,\\
&h_{23} = -(b\alpha\cos\theta - y_d\sin\theta) / x^-_{i,k}(3)^2,\\
&h_{12} = h_{14} = h_{21} = h_{22} = h_{24} = 0,
\end{aligned}
\tag{3}
$$

where $z_{j,k}$ denotes the j^{th} measurement vector that contains x_{dl} and d_{acc}. A measurement corresponding to the track is selected in the data association step. In the update step, a *posterior* state vector ($\hat{x}^+_{i,k}$) of the i^{th} track is recursively updated with the associated j^{th} corresponding measurement ($z_{j,k}$).

$$
\begin{aligned}
&K_{i,k} = \hat{P}^-_{i,k}H^T_{j,k}(H_{j,k}\hat{P}^-_{i,k}H^T_{j,k} + R_k)^{-1},\\
&\hat{x}^+_{i,k} = \hat{x}^-_{i,k} + K_{i,k}(z_{j,k} - h_k(\hat{x}^-_{i,k})),\\
&\hat{P}^+_{i,k} = (I - K_{i,k}H_{j,k})\hat{P}^-_{i,k},
\end{aligned}
\tag{4}
$$

where $K_{i,k}$ denotes the Kalman gain of the i^{th} track.

3.2 3D feature-based tracking

The 3D feature-based visual tracking module consists of feature extraction, feature tracking, feature selection, 3D feature clustering, model selection and ROI estimation, as shown in Fig. 3. A feature from an accelerated segment test (FAST) detector [28] is used to extract distinctive features due to its speed and high repeatability. The FAST detector classifies a point as a corner feature if n contiguous pixels exist in the circle of the feature. The n pixels should all be brighter or all darker than the intensity of the point. Each of the 16 neighbourhood pixels in the circle have one of three states in the circle. The states are represented by darker (d), brighter (b), and similar (s) pixels. The KLT tracker localizes the correspondences of the features extracted from the previous image. A feature selection procedure is essential in removing the erroneous corresponding features pairs in illumination and appearance changes. Incorrectly matched features are removed using the 2D and 3D feature matching schemes. The census transform and Hamming distance are used to measure the similarity of feature pairs [29]. Census transform is determined by the relative order of the local intensity; a binary pattern is measured using the Hamming distance. Therefore, the method is much more robust than the NCC matching method near object boundaries [29]. The matching algorithm is executed in both a 2D grey image and a 3D depth image to remove wrongly estimated feature pairs.

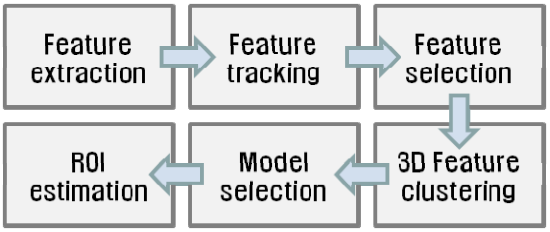


Figure 3. Block diagram of 3D feature-based visual tracking

$$\begin{cases} \text{if}(D_H(T_c(f_{t-1}^i), T_c(f_t^{-i})) > \gamma), & \text{the feature is selected,} \\ \text{else,} & \text{the feature is rejected,} \end{cases} \quad (5)$$

where $T_c(x)$ denotes the census transform function of a feature x, and $D_H(a,b)$ indicates the Hamming distance between the a and b vectors. f_{t-1}^i and denote the i^{th} feature in the t-1 image and the corresponding feature in the t image, respectively. γ is a fixed threshold value for accepting the features. The feature selection is executed in a grey image and a depth image.

One of the problems of using a feature-based tracker is that it is very difficult to select only the features corresponding to the target object. When an object is estimated by a misaligned ROI, there are many more outlier features that correspond to the background or to other objects (Fig. 4). Consequently, the outlier features cause the model parameters to be incorrectly estimated. The 3D feature clustering method deals with the problem while minimizing the number of these outlier features. The features are clustered in 3D global position and motion spaces using the iterative scheme. In 3D global position clustering, the features are projected into 3D global coordinates using the IPM model [24].

Figure 4. Many outlier features in misaligned ROI

$$P_i^g = \begin{bmatrix} X_g \\ Y_g \\ Z_g \end{bmatrix} = \begin{bmatrix} \dfrac{(x_{dl} + x_{dr})\{(Y_g + h)\sin\theta + Z_g\cos\theta\}}{2\alpha} \\ \dfrac{by_d\cos\theta + \alpha b\sin\theta - d_{int}h}{d_{int}} \\ \dfrac{b(\alpha\cos\theta - y_d\sin\theta)}{d_{int}} \end{bmatrix}, \quad (6)$$

where P_i^g denotes the 3D global position of the i^{th} feature. X_g, Y_g, and Z_g are feature positions in global coordinates. d_{int} indicates the integer disparity of the feature. The Mahalanobis distance (d_m) is used to reject the outliers.

$$\begin{cases} \text{if}(d_m(P_i^g) > T_p), & \text{the feature is removed,} \\ \text{else,} & \text{the feature is selected,} \end{cases} \quad (7)$$
$$d_m(P_i^g) = (P_i^g - P_m^g)^T \Sigma_P^{-1}(P_i^g - P_m^g),$$

where P_m and Σ_P denote the mean and covariance of the features in 3D global position, respectively. T_p is a threshold value to reject the outlier features. For 3D global motion clustering, we calculate the displacement of selected features in global coordinates.

$$\begin{cases} \text{if}(d_m(M_i) > T_M), & \text{the feature is removed,} \\ \text{else,} & \text{the feature is selected,} \end{cases}$$
$$d_m(M_i) = (M_i - M_m)^T \Sigma_M^{-1}(M_i - M_m), \quad (8)$$
$$M_i = P_i^t - P_i^{t-1},$$

where M_i indicates the motion vector of the i^{th} feature in 3D global coordinates, M_m and Σ_P are the mean and covariance of motion vectors in the 3D global coordinates, respectively, and T_M is a threshold value related to the motion vector. The mean and covariance of features are updated with the selected features in each iteration. The features are iteratively selected and rejected until the means of the position and motion of the 3D features

converge. The finally selected features are used to estimate the warping matrix with the RANSAC scheme. The current ROI can be estimated by the transformation matrix and the previous ROI.

3.3 Hierarchical data association

Data association problems were originally addressed using the multiple object tracking problem in radar systems. In recent decades, data association methods have been applied to intelligent vehicles [14] and surveillance fields [15, 26] for multiple object tracking. To solve the assignment problem, an association cost matrix (C) should be calculated using the similarity function. In [14], the similarity function of global distance and the sub-pixel disparity are used to calculate an association cost. Even though, according to the experimental results, track identity switching error is not known to have occurred, a few tracks often link to false detections such as guard rails or side walls. In this study, we enhance the discriminating power while using local position distance and appearance similarity as well as 3D global distance.

$$c(t_i, h_j) = \omega_G f_G(t_i, h_j) + \omega_L f_L(t_i, h_j) + \omega_A f_A(t_i, h_j), \quad (9)$$

where $c(t_i, h_j)$ indicates the association cost value of the i^{th} track and j^{th} hypothesis, and $f_G(t_i, h_j)$, $f_L(t_i, h_j)$ and $f_A(t_i, h_j)$ represent the functions of global distance, local distance, and appearance distance, respectively, of the i^{th} track and j^{th} hypothesis. ω_G, ω_L, and ω_A denote fixed weighting factors and are set at 0.5, 0.3, and 0.2, respectively.

Global distance function is represented by the global and disparity distance between the track's prediction and the measurement.

$$f_G(t_i, h_j) = \exp\left[-\sqrt{(t_i^d - h_j^d)^2 + (t_i^x - h_j^x)^2 + ((t_i^z - h_j^z)/t_i^z)^2}\right], \quad (10)$$

where t_i^d, t_i^x, and t_i^z denote disparity, lateral distance, and longitudinal distance of the i^{th} track's prediction, and h_j^d, h_j^x, and h_j^z represent disparity, lateral distance, and longitudinal distance of the j^{th} hypothesis, respectively. The difference rate of the longitudinal distance is used instead of the longitudinal distance difference because distance errors increase exponentially as the distance increases.

Local distance is computed using the overlap ratio (δ_o) and normalized distance of the centre point between the ROIs.

$$f_L(t_i, h_j) = \exp\left[\delta_o - \sqrt{\frac{\left(u_{t_i} - u_{h_j}\right)^2 + \left(v_{t_i} - v_{h_j}\right)^2}{w_{t_i}^2 + h_{t_i}^2}} - 1\right], \quad (11)$$

$$\delta_o = \frac{R_{t_i} \cap R_{h_j}}{R_{t_i} \cup R_{h_j}}$$

where u_{t_i} and v_{t_i} denote the centre of the predicted ROI of the i^{th} track; u_{h_j} and v_{h_j} denote the centre of the ROI of the j^{h} hypothesis. w_{t_i} and h_{t_i} denote the width and height of the predicted ROI of i^{th} track. R_{t_i} and R_{h_j} indicate the regions of the i^{th} track and j^{h} hypothesis, respectively. Local distance helps to prevent a tracking vehicle from connecting to false detections or other vehicles around the target vehicle.

The histogram of gradient (HOG) [30] for the appearance similarity function distinguishes a correct hypothesis from an incorrect hypothesis.

$$f_A(t_i, h_j) = \exp\left[-\lambda d_{Bh}\left(t_i, h_j\right)\right]$$
$$d_{Bh}\left(t_i, h_j\right) = \sqrt{1 - \sum_{u=0}^{m-1} H_{t_i}(u) H_{h_j}(u)}, \quad (12)$$

where λ denotes a constant value, and $H_{t_i}(u)$ and $H_{h_j}(u)$ indicate the HOGs of the i^{th} track and the j^{th} hypothesis. $d_{Bh}(p,q)$ represents the Bhattacharyya distance between p and q; m is the number of histogram bins [31]. Actually, the HOG does not discriminate between the target vehicle and other vehicles very accurately because all of the vehicles have similar HOG appearances. However, the HOG model is efficient at distinguishing the target vehicle from false detections such as walls and guard rails.

The hierarchical data association method assigns existing current tracks to multiple hypotheses. This method has three stages, which are the track-to-OSH, track-to-GH, and track-to-TSH association. In the track-to-OSH stage, all the existing tracks are assigned to OSH using the GNN data association algorithm. The optimal assignment matrix (A) for one-to-one mapping is determined by

$$A = \arg\max_{\hat{A}} \sum c_{ij} a_{ij}, \quad (13)$$

where c_{ij} indicates the distance between the i^{th} track and the j^{th} hypothesis, and a_{ij} is the assignment value, which becomes 1 or 0. If the i^{th} track and j^{th} hypothesis are associated, the value becomes 1; otherwise, it becomes 0. These values should be mutually exclusive for one-to-one mapping. If the association cost is higher than the validation gate threshold, even when the assignment value is 1, the assignment value becomes 0. The validation gate scheme removes unlikely track-to-OSH pairs. In the track-to-GH association, the unassigned tracks determine their corresponding GH using the nearest neighbourhood method. A strict validation gate using the overlap ratio and the global distance are used to remove the unlikely track-to-GH pairs in this stage. In the Track-to-TSH stage, the remaining tracks link to the TSH, which should exist in the validation gate calculated using the global distance function.

3.4 Particle filter using multiple hypotheses

The Bayesian-based object tracking framework consists of a motion model and an observation model; the target states are estimated by maximum *a posteriori* (MAP) probability.

$$\begin{aligned} X_k &= \arg\max_{\hat{x}_k} p(X_k \mid Z_k) = \frac{p(x_k \mid x_{k-1}) p(z_k \mid x_k)}{p(z_k \mid Z_{k-1})} p(X_{k-1} \mid Z_{k-1}), \\ &\propto p(x_k \mid x_{k-1}) p(z_k \mid x_k) p(X_{k-1} \mid Z_{k-1}), \\ X_k &= [x_0\, x_1 \ldots\ldots x_k],\ Z_k = [z_0\, z_1 \ldots\ldots z_k] \end{aligned} \tag{14}$$

where x_k and z_k indicate a state vector and an observation vector, respectively. In this work, the state vector and the observation vector consist of the horizontal and vertical centres (u, v) and the width (w) and height (h) of the ROI. The uncertainty of the motion is very high in the image plane due to the ego-motion and nonlinear projection. As a result, the Gaussian motion model based on the previous posterior state is used to estimate the prior state of the target. A *priori probability* of the j^{th} sample (x_k^j) is calculated by

$$p(x_k^j \mid x_{k-1}) = \frac{1}{(2\pi)^{k/2} |\Sigma_m|^{1/2}} \exp\left(-\frac{(x_k^j - x_{k-1})^T \Sigma_m^{-1} (x_k^j - x_{k-1})}{2} \right), \tag{15}$$

where x_{k-1} indicates the previous posterior state vector, and Σ_m and k denote the covariance matrix of the Gaussian motion model and the dimension of the state vector, respectively.

In our tracking-by-multiple hypotheses framework, several measurements are used in the observation model; these observations correspond to multiple hypotheses, such as GH, OSH and TSH. All the tracks are initialized from a few of the consecutive associated OSH; they are terminated by a few of the unassociated OSH and GH. GH contains many false detections and poor ROI precision, but provides high detection probability, because the approach extracts all the obstacles on the road regardless of their object class. The state of the tracks that are not linked to the OSH is updated with the associated GH, which allows the track to be maintained for a longer time. TSH is mainly dependent on the previous target state. TSH provides relatively good results in general conditions, but is prone to failing to track the target during abrupt motions or illumination changes. The TSH enables a track to maintain a stable state in the presence of abrupt variations of the GH and OSH.

In the observation model of the tracking-by-multiple hypotheses framework, the likelihood term is calculated by the weighted sum of these noisy multiple hypotheses.

$$\begin{aligned} p(z_k \mid x_k) &= \sum_{i=1}^{n} \mu_i (z_k^i \mid x_k), \\ \mu_i &= \frac{p(z_k^i \mid x_k^-)}{\sum_{i=1}^{n} p(z_k^i \mid x_k^-)}, \\ p(z_k^i \mid x_k^-) &= \eta_L \exp\left(-\frac{(x_k^- - z_k^i)^T \Sigma^{-1} (x_k^- - z_k^i)}{2} \right), \end{aligned} \tag{16}$$

where Σ denotes the covariance matrix of the residual ($x_k^- - z_k^i$) and η_L is the normalization value. The weighting factor (μ_i) is calculated using the conditional probability of each hypothesis given *a priori* state vector (x_k^-). The likelihood of the i^{th} hypothesis (z_k^i) given the j^{th} sample (x_k^j) is calculated by

$$p(z_k^i \mid x_k^j) = \frac{1}{(2\pi)^{k/2} |\Sigma|^{1/2}} \exp\left(-\frac{(x_k^j - z_k^i)^T \Sigma^{-1} (x_k^j - z_k^i)}{2} \right) \cdot \tag{17}$$

A posterior probability of each sample is calculated by the product of the prior probability and the likelihood. A sample with maximum probability is selected and the current target state is updated using the sample state.

$$x_k = \arg\max_{x_k^j} \sum_{i=1}^{n} \mu_i p(x_k^j \mid x_{k-1}) p(z_k^i \mid x_k^j) p(X_{k-1}^j \mid Z_{k-1}). \tag{18}$$

The number (N) of samples per object is set to 1,000. They are used to estimate the optimal target state in a particle filter. The posterior probability density is recursively propagated using the probabilities of the samples at every time step.

4. Experimental results

4.1 Experimental setup

Real-world stereo sequences are captured in various scenarios from stereo cameras to test and verify the performance of our method. All the images are 640 × 352 × 8 bpp at 15 fps from a stereo camera mounted on a moving vehicle with a 0.3 m baseline (Fig. 5). Depth images are obtained by a software program based on the belief propagation algorithm. It is a time-consuming process and the algorithm is implemented in the FPGA system for real-time processing. Our software platform includes obstacle detection, vehicle recognition and multiple vehicle tracking modules (Fig. 1).

Figure 5. Stereo vision system mounted on vehicle

Four different scenarios (Fig. 6) are selected for quantitative evaluation; many more test scenarios are used for qualitative analysis. The four scenes are captured from the following settings: urban roads in heavy traffic, cluttered roads with severe illumination change, urban roads on rainy days and highways with curves. Ground truths for each scenario are manually annotated. The tracking performance is evaluated using a metric that is widely used in the multiple object tracking field [14-15, 26, 32]. Two ground truths are used to count the numbers of false negative and false positive alarms while considering limited distance and occlusion conditions. One is a mandatory ground truth, which represents all the vehicles with full appearance at less than 70 m and includes tracking vehicles that are partially occluded at less than 70 m. The other is an optional ground truth, which includes partially occluded vehicles being initialized, and vehicles at more than 70 m. The vehicle recognition system fails to classify partially occluded vehicles correctly; also, distant vehicles are difficult to recognize due to their small size. The number of false negative alarms is counted when a vehicle with mandatory ground truth is not detected. The number of false positive alarms is counted when the estimated ROI fails to correspond to both the mandatory and the optional ground truths.

	Scenario 1	Scenario 2	Scenario 3	Scenario 4
Number of frames	726	997	557	601
Number of vehicles	2531	1185	1114	1143
Image				

Figure 6. Test datasets for quantitative evaluation

The CLEAR MOT metric [32] gives both the multiple object tracking precision (MOTP) score and the multiple object tracking accuracy (MOTA) score. MOTP indicates a measure for localization precision of the estimated ROI. It is calculated using the intersection ratio over the union of two bounding boxes.

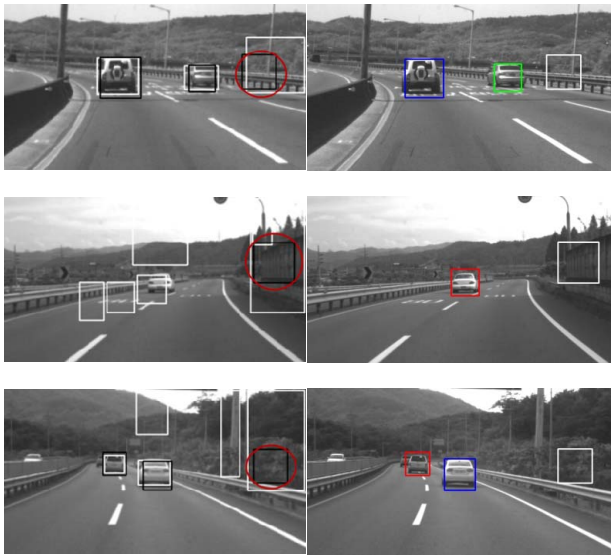
$$\mathrm{MOTP} = \frac{1}{\mathrm{N_g}} \sum \frac{\mathrm{R_g^{k,i}} \cap \mathrm{R_e^{k,i}}}{\mathrm{R_g^{k,i}} \cup \mathrm{R_e^{k,i}}} \times 100\,, \tag{19}$$

where $\mathrm{R_g^{k,i}}$ is the region of the ground truth of the i^{th} vehicle at the k^{th} frame and $\mathrm{R_e^{k,i}}$ denotes the region of the estimated ROI. N_g indicates the total number of ground truths. MOTA provides a measure for the localization accuracy of the estimated ROI. It is evaluated using the sum of missed detections (N_m), false detections (N_f), and track identity switches (N_s).

$$\mathrm{MOTA} = \left(1 - \frac{\mathrm{N_m} + \mathrm{N_f} + \mathrm{N_s}}{\mathrm{N_g}}\right) \times 100\,, \tag{20}$$

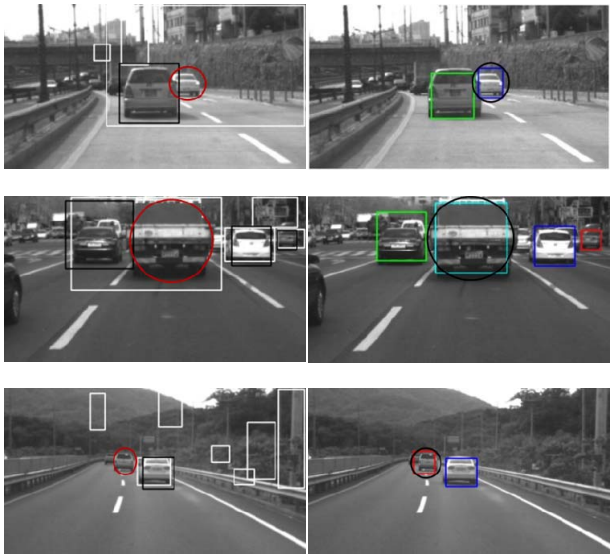
4.2 Evaluation and analysis

We tested and analysed the proposed method qualitatively and quantitatively using several image sequences captured from various real road environments. Most walls, guardrails and trees around roads are extracted from the obstacle detection module, because the obstacle detection algorithm detects all the obstacles on the road. The vehicle recognition module often mistakes these false detections for vehicles due to erroneous vehicle recognition algorithms (Fig. 7(a)). In the multiple vehicle tracking, most false detections are removed during track initialization due to their sparse occurrence (Fig. 7(b)). Vehicle detection misses a partially occluded vehicle (first raw image), one of two vehicles that are close together (second raw image), and a vehicle in the far distance (third raw image), as shown in Fig. 8(a). However, the visual tracking module estimates the missed target ROI using the previous ROI (Fig. 8(b)). The vehicle detection module often gives an unstable ROI state such as a bigger ROI (first raw image), smaller ROI (second raw image), or misaligned ROI (third raw image), as shown in Fig 9(a). The target states are smoothed even though the ROI states are abruptly changed due to noisy vehicle recognition (Fig. 9(b)). A track is terminated if the track is not linked to the corresponding observations for several consecutive frames. In Fig. 10, the tracks are not associated with any vehicle recognition outputs even though the obstacle detection module estimates the ROI of the vehicle correctly. Errors in vehicle recognition often occur in small ROI (first raw image), ROI with a part of a vehicle (second raw image), and ROI in dark lighting conditions (third raw image). Unassigned tracks determine their corresponding GH using hierarchical data association; the tracks can be updated and maintained with the associated GH.



(a) (b)

Figure 7. False detection removal. (a) Vehicle detection results: White boxes and black boxes represent results of obstacle detection and vehicle recognition, respectively. Red circles indicate false detections. (b) Results of multiple vehicle tracking: Colour boxes denote the tracking vehicles and white box indicates that the vehicles are being initialized, which are not regarded as detected vehicles in this frame.



(a) (b)

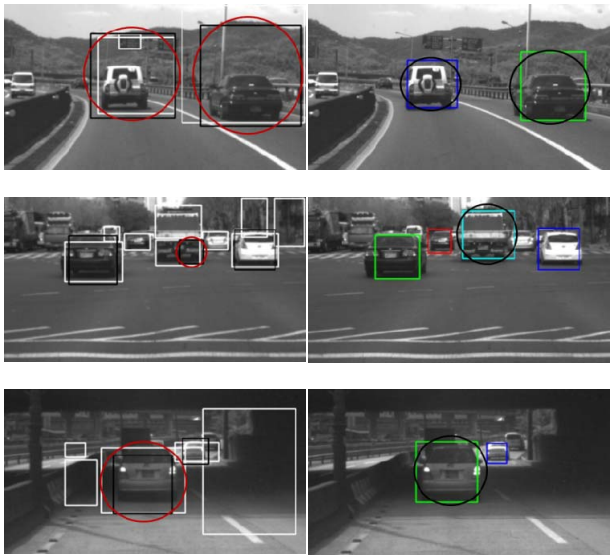
Figure 8. Recovery of the ROI of missed detection. (a) Vehicle detection results: Red circles indicate missed detections. (b) Results of multiple vehicle tracking: Black circles indicate tracked ROIs.

In scenario 1, there are many missed detections when vehicles are close to or occluded by other vehicles for several tens of frames, the track cannot be initialized due to their sparse detection outputs and the number of false negative alarms are increased in this period (Fig. 11). In scenario 2, when the false detections (walls and guard rails) are associated with incorrect tracks for a few consecutive frames, the false detections are propagated using visual tracking, even though the false detections are not detected in subsequent frames (Fig. 12). In some scenarios, there are a few visual tracking errors for far away vehicles in heavy traffic (Fig. 13(a)), vehicles in bad illumination conditions (Fig. 13(b)), and vehicles in noisy images due to raindrops (Fig. 13(c)).

Table 1 shows the quantitative evaluation results for four different real world scenarios. Recall and precision as well as MOTA and MOTP are reported to indirectly compare with other methods. In the tracking-by-multiple hypotheses framework, the target state is estimated by the stochastic particle filter. We executed the method ten times to determine the mean and standard deviations. The experimental results show that the scores of MOTA and MOTP in our proposed method outperformed those in the vehicle detection method in all the test scenarios. In scenario 1, there were many missed detections due to close and occluded vehicles. In scenario 2, false detection propagation errors occurred due to a few consecutive false detections. In scenario 3, there were some errors in the vehicle recognition module due to very noisy images. The recall in the vehicle detection method is very low. However, the obstacle detection module can detect many vehicles, and the tracking-by-multiple hypotheses framework can update and maintain the target state with the GH (Fig. 14). In scenario 4, a track was not initialized when vehicles were occluded by other vehicles for dozens of frames, and most of the false negative alarms occurred in this period. Our videos for experimental results are available on YouTube [33-36]. In our future research, we will adopt a more advanced object detection method and will show the effectiveness of the proposed approach using the object recognition with the obstacle detection.

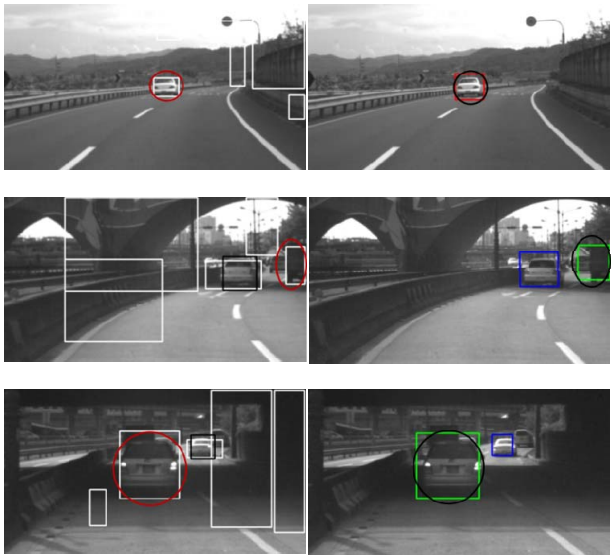
		Recall	Precision	MOTA	MOTP
S1	Vehicle detection	70.7%	97.2%	68.7%	66.3%
	Proposed method	**89.7±0.5%**	**99.0±0.5%**	**88.9±0.6%**	**67.1±0.6%**
S2	Vehicle detection	77.8%	80.1%	58.5%	66.2%
	Proposed method	**95.2±0.5%**	**96.3±0.5%**	**91.5±0.9%**	**67.9±0.4%**
S3	Vehicle detection	56.3%	94.4%	53.0%	68.4%
	Proposed method	**99.4±0.8%**	**99.4±0.8%**	**98.8±1.6%**	**68.4±1.1%**
S4	Vehicle detection	79.6%	90.5%	71.1%	65.6%
	Proposed method	**97.3±0.1%**	**99.5±1.1%**	**96.7±1.1%**	**74.0±0.4%**

Table 1. Quantitative evaluation results



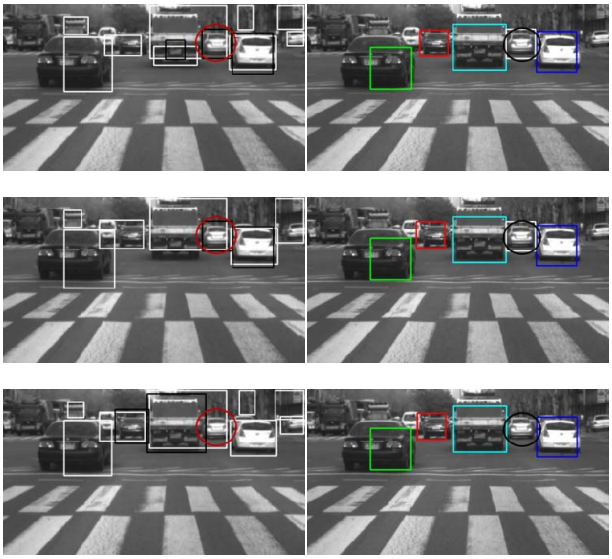
(a) (b)

Figure 9. Smoothness of unstable ROI. (a) Vehicle detection results: Red circles indicate the misaligned ROI. (b) Results of multiple vehicle tracking: Black circles indicate updated ROIs.



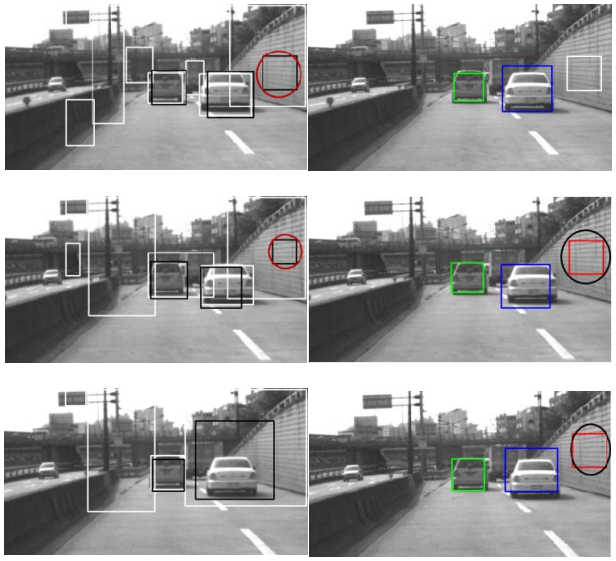
(a) (b)

Figure 10. Track-to-GH association for track maintenance. (a) Vehicle detection results: Red circles indicate the ROI of GH. (b) Results of multiple vehicle tracking: Black circles indicate the track states are updated with the GH.



(a) (b)

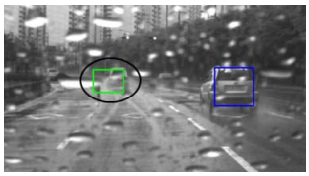
Figure 11. Track initialization failure. (a) Vehicle detection results: Red circles indicate two consecutive vehicles are detected, but the vehicle is not detected in the third image. (b) Results of multiple vehicle tracking : White box indicates a track-initializing vehicle. Black circle indicates track initialization failure due to deficiency of consecutive detections.



(a) (b)

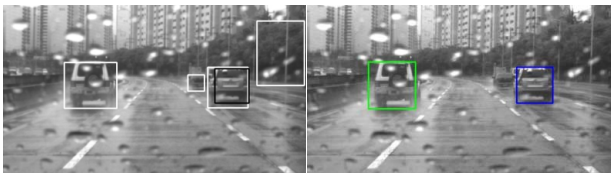
Figure 12. False detection propagation error caused by visual tracking. (a) Vehicle detection results: Red circles indicate false detections. (b) Results of multiple vehicle tracking: Black circle indicates false detection propagation error.

(a) (b)



(c)

Figure 13. Visual tracking errors (a) Far away vehicle in heavy traffic. (b) Vehicle in bad illumination condition (c) Vehicle in noisy image due to raindrops.



(a) (b)

Figure 14. (a) Missed detection in vehicle recognition. (b) Updated ROI with GH in tracking-by-multiple hypotheses framework.

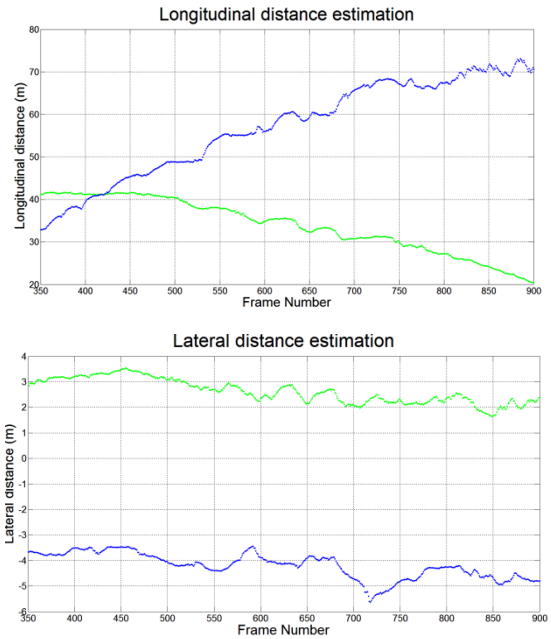


Figure 15. Longitudinal and lateral distance estimation for two target vehicles in scene 3.

Fig. 15 shows the vehicle trajectories of longitudinal distance and lateral distance for scenario 3. The experimental results verify that our method can estimate the trajectories of target vehicles reliably even though noisy stereo images were captured on a rainy day.

All the software algorithms were implemented in Visual C++ using OpenCV 2.2 on a PC platform with a quad core 2.83 GHz CPU. The values of the parameters used for the experiments are summarized in Table 2. The frame rate of all the software algorithms, such as obstacle detection, vehicle recognition and multiple vehicle tracking, is about 10 to 15 frames per second. The frame rate of the multiple vehicle tracking algorithm is about 15 to 19 frames per second. The processing time for all the test scenes is described in Table 3.

	parameters	values
Global position tracking	η_x	1 m/s²
	η_z	5 m/s²
	σ_x	1.0
	σ_d	0.5
Visual tracking	γ	0.7
	T_p	9.0
	T_M	9.0
Data association	ω_G	0.5
	ω_L	0.3
	ω_A	0.2
Particle filter	N	1,000

Table 2. Values of parameters used in our experiments

	Obstacle detection	Vehicle recognition	**Multiple vehicle tracking**	Total
S1	8.7±1.7 ms	10.4±2.6 ms	**68.9±5.7 ms**	88.0±7.3 ms
S2	8.3±1.2 ms	12.3±2.3 ms	**56.4±4.4 ms**	77.1±5.1 ms
S3	7.8±0.8 ms	8.7±1.9 ms	**54.6±3.4 ms**	71.2±4.6 ms
S4	7.6±0.8 ms	7.1±2.0 ms	**53.6±3.6 ms**	68.3±5.9 ms

Table 3. Processing time per frame

5. Conclusions

In this paper, we proposed a tracking-by-multiple hypotheses framework to improve multiple object tracking accuracy and precision. Most false detections are removed during track initialization; also, the number of missed detections is minimized using 3D visual tracking. A hierarchical data association method was proposed to assign multiple tracks to multiple hypotheses. The particle filter updates the target state using the motion model and the observation model with the multiple associated hypotheses. Experimental results using challenging test scenarios demonstrate that the scores of both MOTA and MOTP are remarkably improved when the results of proposed method and those of the vehicle detection method were compared. Irregular detections caused by occluded vehicles prevent a track from being initialized; false detections propagation errors occur due to visual tracking when the track is initialized by consecutive false

detections. We will work with the track management method to solve these problems. Also, the software algorithm will be optimized and the processing time will be improved using a parallel programming scheme.

6. Acknowledgement

This work was supported by the DGIST R&D Program of the Ministry of Education, Science and Technology of Korea

7. References

[1] Michael S. D, Paul E. R, Christopher B, Chris U (2009) Obstacle detection and tracking for the urban challenge. IEEE Transactions on Intelligent Transportation System 10(3):475-485.

[2] Sun Z, Bebis G, Miller R (2006) On-road vehicle detection: a review. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(5):694 – 711.

[3] Sakai H, Suzuki Y, Takagi K, Morikawa K (2011) Pedestrian detection and tracking using in-vehicle Lidar for automotive application. Proceedings of IEEE Intelligent Vehicles Symposium 734-739.

[4] Lundquist C, Orguner U, Schon T. B (2009) Tracking stationary extended objects for road mapping using radar measurements. Proceedings of IEEE Intelligent Vehicles Symposium 405-410.

[5] Chang J. –Y, Cho C. –W (2006) Vision-based front vehicle detection and its distance estimation. Proceedings of IEEE International Conference on Systems, Man and Cybernetics 3:2063 – 2068.

[6] Wu B. –F, Chen W. –H, Chang C. –W, Chen C. –J (2007) A new vehicle detection with distance estimation for lane change warning systems. Proceedings of IEEE Intelligent Vehicles Symposium 698-703.

[7] Kowsari T, Beauchemin S. S, Cho J (2011) Real-time vehicle detection and tracking using stereo vision and multi-view AdaBoost. Proceedings of IEEE Conference on Intelligent Transportation System 1255-1260.

[8] Bajracharya M, Moghaddam B, Howard A, Brennan S, Matthies H. L (2009) A fast stereo-based system for detecting and tracking pedestrians from a moving vehicle. International Journal of Robotics Research 28(11-12):1466-1485.

[9] Barth A, Franke U (2009) Estimating the driving state of oncoming vehicles from a moving platform using stereo vision," IEEE Transactions on Intelligent Transportation System 10(4):560-571.

[10] Nedevschi S, Bota S, Tomiuc C (2009) Stereo-based pedestrian detection for collision-avoidance applications. IEEE Transactions on Intelligent Transportation System 10(3):380-391.

[11] Thien-Nghia N, Michaelis B, Al-Hamadi A, Tornow M, Meinecke M (2012) Stereo-camera-based urban environment perception using occupancy grid and object tracking. IEEE Transactions on Intelligent Transportation System 13(1):154-165.

[12] Geronimo D, Lopez A. M, Sappa A. D (2010) Survey of Pedestrian Detection for Advanced Driver Assistance Systems. IEEE Transactions on Pattern Analysis and Machine Intelligence 32(7):1239-1258.

[13] Caraffi C, Vojir T, Trefny J, Sochman J, Matas J (2012) A system for real-time detection and tracking of vehicles from a single car-mounted camera. Proceedings of IEEE Intelligent Transportation Systems:975-982.

[14] Lim Y. -C, Lee M, Lee C. -H, Kwon S, Lee J.-H (2011) Integrated position and motion tracking method for online multi-vehicle tracking-by-detection. Optical Engineering 50(07):077203.

[15] Breitenstein M. D, Reichlin F, Leibe B, Koller M. E, Van G. L (2011) Online multiperson tracking-by-detection from a single, uncalibrated camera. IEEE Transactions on Pattern Analysis and Machine Intelligence 33(9):1820-1833.

[16] Sochman J, Matas J (2005) WaldBoost - Learning for Time Constrained Sequential Detection. Proceedings of IEEE Conference Computer Vision and Pattern Recognition:150–157.

[17] Kalal Z, Matas J, Mikolajczyk K (2010) P-N Learning: Bootstrapping Binary Classifiers by Structural Constraints. Proceedings of IEEE Computer Vision and Pattern Recognition:49-56.

[18] Doulam A (2010) Online Dynamic tracking re-adjustment: a method for automatic tracking recovery in complex visual environments. Multimedia Tools and Applications 50(1):49-73.

[19] Felzenszwalb P. F, Huttenlocher D. P (2006) Efficient belief propagation for early vision. International Journal of Computer Vision 70(1):41-54.

[20] Labayrade R, Aubert D, Tarel J. –P (2002) Real time obstacle detection in stereovision on non flat road geometry through "v-disparity" representation," Proceedings of IEEE Intelligent Vehicles Symposium 646-651.

[21] Lee C. -H, Lim Y. -C, Kwon S, Lee J. –H (2011) Stereo vision-based vehicle detection using a road feature and disparity histogram," Optical Engineering, 50(2):027004.

[22] Viola P, Jones M. J (2004) Robust real-time face detection. International Journal of Computer Vision 57(2):137-154.

[23] Kim J. -H, Lee C. -H, Lim Y. -C, Kwon S (2011) Stereo vision-based improving cascade classifier learning for vehicle detection. Lecture Notes in Computer Science 6939:387-397.

[24] Lim Y. -C, Lee M, Lee C. -H, Kwon S, Lee J. -H (2010) Improvement of stereo vision-based position and velocity estimation and tracking using a stripe-based disparity estimation and inverse perspective map-based extended Kalman filter. Optics and Lasers Engineering 48(9):859-868.

[25] Lucas B. D, Kanade T (1981) An iterative image registration technique with an application to stereo vision. Proceedings of International Joint Conference on Artificial Intelligence 674-679.
[26] Benfold B, Reid I (2011) Stable multi-target tracking in real-time surveillance video. Proceedings of IEEE Computer Vision and Pattern Recognition 3457-3464.
[27] Madasu V. K, Hanmandlu M (2010) Estimation of vehicle speed by motion tracking on image sequences. Proceedings of IEEE Intelligent Vehicles Symposium 185-190.
[28] Rosten E, Drummond T (2006) Machine learning for high-speed corner detection. Proceedings of European Conference on Computer Vision 430-443.
[29] Zabih R, Woodfill J (1994) Non-parametric local transforms for computing visual correspondence. Proceedings of European Conference on Computer Vision 151-158.
[30] Dalal N, Triggs B (2005) histogram of oriented gradients for human detection. Proceedings of IEEE Conference Computer Vision and Pattern Recognition 886-893.
[31] Comaniciu D, Ramesh V, Meer P (2003) Kernel-based object tracking. Proceedings of IEEE Transactions on Pattern Analysis and Machine Intelligence 25(5):564 – 577.
[32] Bernardin K, Stiefelhagen R (2008) Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics. EURASIP Journal on Image and Video Processing. 1-10.
[33] https://www.youtube.com/watch?v=M2eQ1dSMjEA.
[34] https://www.youtube.com/watch?v=6xp12lNwXNc.
[35] https://www.youtube.com/watch?v=DpQ72KRlpEo.
[36] https://www.youtube.com/watch?v=JQFnvPtEu3c.