Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 8153474, 15 pages
https://doi.org/10.1155/2021/8153474

Research Article

Moving Camera-Based Object Tracking Using Adaptive Ground Plane Estimation and Constrained Multiple Kernels

Tao Liu [1,2,3] **and Yong Liu** [1,2,3]

[1] *Beijing Key Laboratory of Network System Architecture and Convergence, Beijing University of Post and Telecommunications, Beijing, 100876, China*
[2] *Beijing Laboratory of Advanced Information Networks, Beijing University of Post and Telecommunications, Beijing 100876, China*
[3] *School of Information and Communication Engineering, Beijing University of Post and Telecommunications, Beijing 100876, China*

Correspondence should be addressed to Tao Liu; tony6@uw.edu

Received 22 April 2021; Revised 7 July 2021; Accepted 13 July 2021; Published 21 July 2021

Academic Editor: Wen Liu

Moving camera-based object tracking method for the intelligent transportation system (ITS) has drawn increasing attention. The unpredictability of driving environments and noise from the camera calibration, however, make conventional ground plane estimation unreliable and adversely affecting the tracking result. In this paper, we propose an object tracking system using an adaptive ground plane estimation algorithm, facilitated with constrained multiple kernel (CMK) tracking and Kalman filtering, to continuously update the location of moving objects. The proposed algorithm takes advantage of the structure from motion (SfM) to estimate the pose of moving camera, and then the estimated camera's yaw angle is used as a feedback to improve the accuracy of the ground plane estimation. To further robustly and efficiently tracking objects under occlusion, the constrained multiple kernel tracking technique is adopted in the proposed system to track moving objects in 3D space (depth). The proposed system is evaluated on several challenging datasets, and the experimental results show the favorable performance, which not only can efficiently track on-road objects in a dashcam equipped on a free-moving vehicle but also can well handle occlusion in the tracking.

1. Introduction

Currently, video-based traffic surveillance plays an important role in intelligent transportation systems (ITSs). And as more and more people use the dashcam during driving, video analysis based on dascam has thus become a very important research area, and object tracking such as pedestrians and vehicles is a crucial and unavoidable task in this field. By tracking pedestrians or vehicles, their movement trajectories can be collected in the video for advanced analysis, such as human or vehicle flow estimation, collision avoidance of abnormal behavior, and criminal tracking. Therefore, researchers are motivated to develop an effective tracking system, which not only can track objects in the scene but also is able to collect the information for higher-level analysis.

Tracking vehicle and pedestrian in moving cameras is quite challenging due to several reasons. First, the appearance of these objects may change greatly due to nonrigid deformation, different viewing perspectives, and other visual attributes. Second, frequent occlusion by other objects in the scene will cause severe identity switches. Last but not least, object tracking in moving camera is more challenging than that in static cameras, because of the combined effects of rapidly changing lighting conditions, blur, and the issues mentioned above. Moreover, many robust and effective object tracking techniques used in static cameras cannot be directly applied in moving camera, such as background subtraction and constant ground plane assumption, thus making the problem more difficult. Unlike using background-based methods to extract moving objects blobs under static cameras, object detection is widely used in video

analysis under moving camera. Therefore, the challenge becomes to successfully detect objects in the moving cameras and then apply tracking techniques to track the detected ones, which are so-called tracking-by-detection schemes. However, when the object is partially or fully occluded, the detection cannot work well and thus affect the tracking result. Hence, the constrained multiple kernel (CMK) tracking technique was further adopted in the proposed system and facilitated with the estimated ground plane and Kalman filter, to overcome the occlusion issue during the tracking.

In this paper, we extend our previous work [1] and propose an efficient and robust 3D object tracking system based on adaptive ground plane estimation, which also successfully integrates structure from motion (SfM), object detection, CMK tracking, and Kalman filter framework. The proposed system begins with object detection and structure from motion for estimating camera pose. Then, the adaptive ground planes are estimated based on the camera motions, and the 3D location of the objects relative to the cameras can be inferred. By taking 3D information into account, the CMK tracking method is used to overcome the occlusion issue during the tracking. Hence, the proposed system can not only handle the occlusion but also estimate a reliable ground plane simultaneously. Figure 1 shows an example of the tracked objects on the estimated ground plane (the red squares on the ground). The number above the bounding box represents the distance of the detected objects from the camera.

The remaining of this paper is organized as follows: Section 2 gives a brief survey on the related work. In Section 3, we describe the proposed tracking system. The depth CMK tracking which includes depth map construction, CMK tracking, hypothesized association, and Kalman filter are described in Section 4, and Section 5 depicts the adaptive ground plane estimation algorithm. The experimental results are demonstrated in Section 6. Finally, the conclusion of this work is given in Section 7.

2. Related Work

Recently, ground plane estimation-based tracking methods [2–6] have attracted a lot of attention. By applying the ground plane estimation method to each frame of a video sequence for detecting a reliable ground plane, the relative 3D location of the camera and the objects can be inferred, thereby making the object tracking more robust.

In general, the existing ground plane estimation approaches can be roughly divided into two categories: 2D or 3D approaches based on the sensor type. Within 2D approaches, homography is the most popular approach for ground plane estimation, which based on feature correspondence to calculate every pair of consecutive frames and the first requisite is to find a set of reliable feature points lying on the ground plane. Usually, corner detectors such as Harris are used to extract features, followed by a robust estimation technique in which the dominant homography is estimated. Arróspide et al. [7] used Kalman filtering and Conrad and DeSouza [8] used modified expectation maximization to build confidence in the ground plane transformation across successive frames. Both of the two methods assumed the camera can only see the ground plane with objects above it, and the roll angle of sensors is zero. With the homography decomposition results combined with contour searching [9] or a Bayes filter [10] to estimate the ground plane in 2D images, homography has also been successfully used as a first step. However, again the ground plane is assumed to be the area in front of the camera, or the single color ground plane is assumed to occupy the majority of the FOV. The other 2D approaches used depth-image data or histogram of the disparity map [11] instead of traditional RGB image data [12, 13], and Jin et al. [14] proposed a ground plane detection method based on depth map driven, which grows a plane from the largest area having similar depth values in the depth map, and the largest plane is considered to be the ground plane. Kircali and Tek [15] estimated the ground plane by comparing the depth map of new coming frame with a precalibrated depth map in which the ground plane was predefined. Skulimowski et al. [16] used the gradient of the V-disparity pixel values to detect ground plane which has an arbitrary camera roll angle. Furthermore, Cherian et al. [6] reconstruct the depth map from a single RGB image by applying multiple texture-based filters with a Markov random field and estimate the ground plane based on texture-based searching segmentation. Due to the intrinsic features of the algorithm, this approach assumed the ground plane has a unique texture and the camera is parallel to the ground plane. Dragon et al. [17, 18] formulate the ground plane estimation problem as a hidden Markov model (HMM) based on temporal sampling and decomposing of homography. The decomposition of the homography with the highest probability indicates the orientation and ego motion of the camera's movement. Man et al. [19] develop a ground plane estimation approach based on monocular images with a predefined region of interest, which requires a known pitch angle of the camera.

The ground plane estimation method in 3D commonly utilizes the depth sensors as LIDAR [20] or TOF [21] to get the 3D point cloud data, which can provide the 3D structure of the environment and then be used as an effective way to estimate the ground plane. Borrmann et al. [22] use all points of 3D point cloud to calculate, which has high computation cost. RANSAC-like approaches [23, 24], which can then be used as an effective way to estimate the ground plane, are unlimited to number of iteration. Thus, processing time cannot be guaranteed. A less expensive alternative to

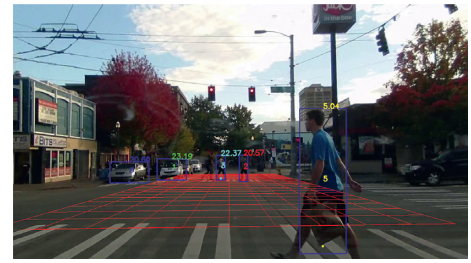


Figure 1: The ground plane estimation and 3D tracking of pedestrians and vehicles based on our system.

generate 3D point clouds is the use of a stereo camera in which the ground plane can be estimated from disparity [25]. Assuming that the scene is static, monocular approaches for simultaneous localizing and mapping (SLAM) can also be used to extract the 3D shape and then the ground plane can be estimated [26, 27]. Zhang and Czarnuch [28] proposed a perspective ground plane estimation approach which combines the robustness of 2D and 3D data analysis. Other 3D approaches [29–31] use the 3D normal vector for each raw data point rather than estimation of the raw points directly. However, we assume that the camera roll and pitch angles are zero. More recently, machine learning technique has been used in ground plane estimation, which requires minimal orientation variations (i.e., $0 \sim 15^{\circ}$) [32].

Although the above approaches can successfully detect the ground plane and achieve good experimental results, they are specifically designed to only produce one single ground plane based on the available data and not suitable for the unpredictability of dynamic road conditions. In addition, these approaches do not utilize the estimated camera pose information. In addition, the camera's pose is the most significant factor for representing the ground plane in the scene. The reliability and accuracy of the ground plane estimation can thus be improved by taking advantage of the camera pose information.

Our proposed tracking system is inspired by the approach in [33], which also has mounted the monocular dashcam on a free-moving vehicle. However, due to the driving road condition is continuously changing, if the ground plane is only estimated in the beginning may not be applicable for the entire video sequence, therefore, it is very useful to take advantage of the camera's pose information estimated from the essential matrix calculation phase. In contrast to the most existing ground plane estimation methods, our approach introduces the estimated camera yaw angle as a feedback to estimate ground plane adaptively, which aims to overcome the deficiency of the previous methods caused by fixed frame window for smoothing the results. Based on the reliably estimated ground plane, we can locate the detected objects in 3D space and combine CMK tracking with the 3D information, so as to deal with the partial or fully occlusion issues during tracking.

3. Overview of the Proposed System

The proposed tracking system is shown in Figure 2. After converting the video from the dashcam to image sequences, there are two parallel procedures launched simultaneously. In the structure from motion phase, the proposed system extracts the Harris corner features in the current image at time step t and matches them to the features observed in the previous N frames. By using the singular value decomposition (SVD), we can estimate the camera's essential matrix for each image frame. Then, according to the camera essential matrix, the ground plane for the entire image sequences can thus be estimated adaptively, where we assume the dashcam is mounted on the vehicle with a fixed height. Meanwhile, a pretrained object detector is adopted to detect desired objects such as vehicle and pedestrian in the image sequences. In the pose estimation stage, the 2D locations of detected objects can be back-projected to 3D locations by using the estimated ground plane. Once the 3D locations of the detected objects is obtained from the pose estimation stage, the depth CMK tracking is applied to track them in the Kalman filter framework. First, for each target, the 3D locations of its candidate are predicted by the Kalman filter predication. Then, the CMK tracking is applied to relocate the candidate's 3D locations by maximizing the similarity between candidates and target. The Kalman filter continually updates and finally gets the reliable tracking result. Besides, based on the object's 3D information relative to the camera motion, a depth map can be constructed to represent the relative 3D locations of all the detected objects. Therefore, with the help of depth information between the targets, the proposed system not only is able to effectively track objects but also can overcome occlusion during the tracking.

3.1. Robust Feature Extraction. The ideal ground plane estimation largely depends on the selected image feature detector, which should contain the invariance of rotation, scale, and image noise. Scale-invariant feature transform (SIFT) [34] feature is a very effective scale-space feature, but it can be very time-consuming for real-time applications. As for the speeded-up robust features (SURFs) with lower computational complexity, its stability is a major problem because it often detects unstable features even after edge suppression as a post treatment. The Harris corner feature detector is thus introduced to solve the above issues, which has also been widely studied in the previous works [35–38]. Firstly, its feature extraction execution speed can be used in real-time applications with reasonable robustness in accuracy. Secondly, to robustly estimate the ground plane, more corner points on the ground plane are welcome to participate in the calculation of the camera parameters. Figure 3 shows an example of using the Harris corner feature detector to extract feature points. The detected feature points in the current image are marked with green crosses. Feature points that are detected as outliers during the processing are marked with red crosses. These points can be matched from one image frame to the next by choosing matches that have the highest cross-correlation of image intensity for regions surrounding the points. The paths of the feature points are drawn in orange here.

3.2. Essential Matrix Calculation. Camera pose plays a crucial role in the ground plane estimation for the entire image sequences, and the computation of the camera yaw angle θ is the key to calculate camera pose. According to the study in [39], there are three camera parameters used to describe two relative poses of a camera moving on a planar surface, i.e., the polar coordinates (ρ, φ_c) and yaw angle θ of the second position c_2 relative to the first position c_1 (see Figure 4).

In addition, we can set $\rho = \nu \cdot \Delta t$, where ν is the velocity of the vehicle and Δt is the transition time between the two end positions c_1 and c_2. Therefore, only two parameters (φ, θ) need to be calculated. In addition, according to the

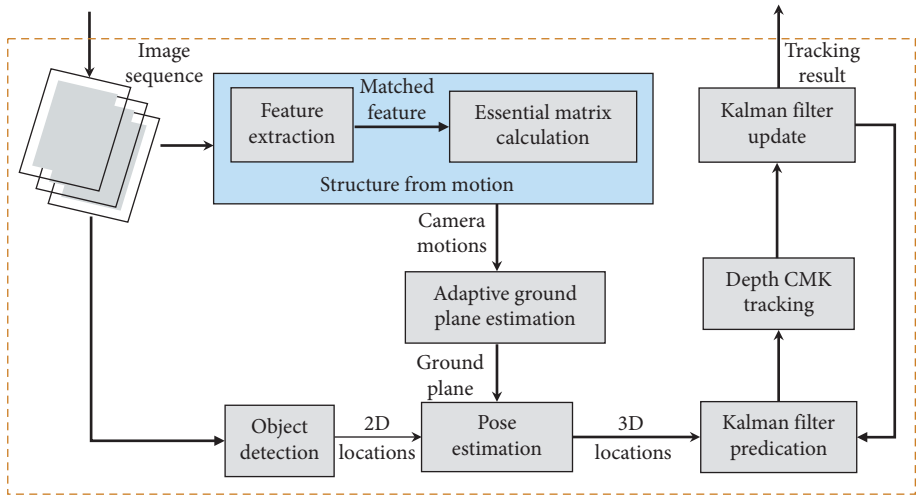


Figure 2: Overview of the proposed system.

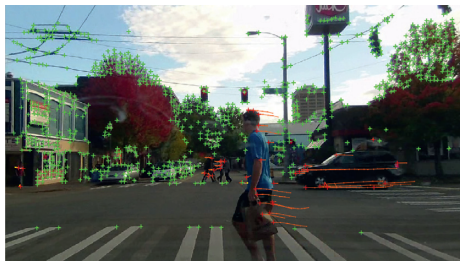


Figure 3: Example of Harris corner feature point extraction.

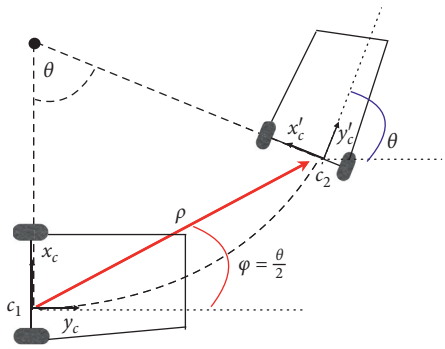


Figure 4: Rotation between camera axes in circular motion.

Ackermann steering principle, a circular motion called the instantaneous center of rotation (ICR) can be used to describe the motion of a camera mounted on a vehicle. The linear driving can be represented along with a circle of infinite radius. With this assumption, we can easily get $\varphi = \theta/2$. Thus, there is only one parameter, and the camera yaw angle θ needs to be calculated.

As we all know, the essential matrix can be represented by the rotation matrix R and the translation matrix T, which are related to the camera pose. Then, we have

$$R = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad T = \rho \cdot \begin{bmatrix} \cos\varphi \\ \sin\varphi \\ 0 \end{bmatrix}, \tag{1}$$

where we consider the camera moves on the (x, y) plane and rotates around the z axis. Given two coplanar points, p and p', which are represented as $p = [\, x \;\; y \;\; z \,]^T$ and $p' = [\, x' \;\; y' \;\; z' \,]^T$ in the image coordinates, they must meet the epipolar constraint:

$$p'^T EP = 0, \tag{2}$$

where E is the essential matrix defined as $E = [T]_\times R$. Note that R is the rotation matrix defined in (1) and $[T]_\times$ denotes the skew symmetric matrix:

$$[T]_\times = \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_X & 0 \end{bmatrix}. \tag{3}$$

Then, using the constraint $\varphi = \theta/2$ and equations (1) and (3), we can obtain the expression of the essential matrix of the camera moving on a planar surface:

$$E = \rho \cdot \begin{bmatrix} 0 & 0 & \sin\frac{\theta}{2} \\ 0 & 0 & -\cos\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} & 0 \end{bmatrix}. \tag{4}$$

By replacing (4) into (2), we can notice that every image points contribute the following homogeneous equation:

$$\sin\frac{\theta}{2}\cdot\left(x'z+z'x\right)+\cos\frac{\theta}{2}\cdot\left(y'z-z'y\right)=0. \tag{5}$$

The rotation angle θ between a pair of successive images can be obtained from (5) as

$$\theta=-2\arctan\left(\frac{y'z-z'y}{x'z+z'x}\right). \tag{6}$$

Conversely, given m consecutive image points, θ can be estimated indirectly by solving linearly for the vector $[\sin(\theta/2),\ \cos(\theta/2)]$ using SVD. To this end, a $m\times 2$ data matrix D is first formed, where each row is formed by the two coefficients of equation (5), as follows:

$$\left[\left(x'z+z'x\right),\left(y'z-z'y\right)\right]. \tag{7}$$

Then, the matrix D is decomposed by using SVD:

$$D_{m\times 2}=U_{m\times 2}\Lambda_{2\times 2}V_{2\times 2}, \tag{8}$$

where the columns of $V_{2\times 2}$ contain the eigenvectors e_i of D^TD. And the eigenvector $e^*=[\sin(\theta/2),\ \cos(\theta/2)]$ corresponding to the minimum eigenvalue minimizes the sum of squares of the residuals, subject to $\|e^*\|=1$. Finally, the yaw angle of the camera θ can be estimated from e^*.

3.3. Object Detection. Object detection is the first step in the tracking-by-detection schemes, and accurate object detection can roughly determine the quality of the tracking system. Unlike detecting objects under static camera, object detection under moving cameras is more challenging due to the dynamic background, illumination changes, and so on. Because the background is constantly changing, the method based on background extraction is no longer applicable for mobile cameras. Therefore, the pretrained object detectors are widely studied in recent years. The work in [40] proposes a human detector by using histogram of gradient (HOG) as the features, which can effectively represent the shape of human. The deformable part model (DPM) [41] extends the concept of [40], which uses a root and several part templates to describe different partitions of the object, and the part templates are spatially connected with the root template according to the predefined geometry, thereby accurately depicting the object. In the latest research, the convolution neural network (CNN)-based object detector has drawn increasing attention and has achieved favorable performance, which can detect hundreds of objects with a high detection accuracy.

In this paper, the objects to be detected and tracked are mainly focusing on the pedestrians and vehicles, which should move on the estimated ground plane. In fact, these objects can be any objects on the road, such as bicycles and animals. In order to avoid detecting other false objects in the field of view, we adopt the state-of-the-art pretrained YOLOv3 detector [42], which uses the most advanced CNN technology to help detecting pedestrians and vehicles. The detector can be embedded independently in the proposed system, so as to functionally perform object detection. To efficiently track the object, the tracking procedure is launched only when the object has been detected in five consecutive image frames; otherwise, the detection is considered as a false alarm. Furthermore, the detected objects are refined by morphological operations to accurately locate their positions.

4. Depth CMK Tracking

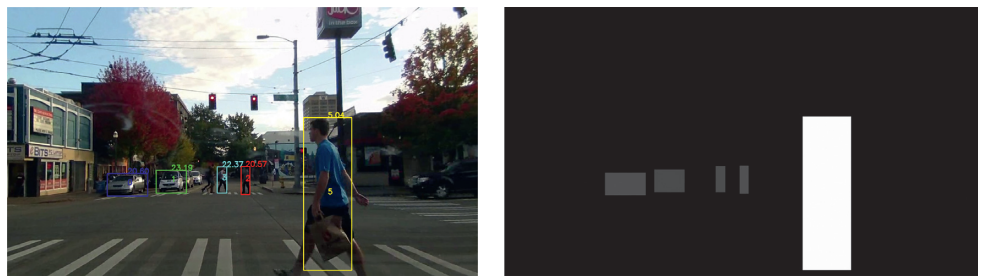
In this section, we mainly describe how to track objects with constrained multiple kernels (CMKs) in 3D space under the framework of the Kalman filter. The depth CMK tracking is triggered to track the objects when its 3D locations are obtained from the pose estimation stage (see Figure 2). In other words, we associate the objects in the current frame with the detected objects in the next frame facilitated with the Kalman filtering. On the other hand, with the help of the depth information, we can get the relative 3D locations between the objects to overcome the occlusion in the tracking. By effectively combining depth information and CMK tracking into the Kalman filter framework, the proposed system can not only track objects effectively but also well handle occlusion problems during tracking.

4.1. Depth Map Construction. A depth map can be constructed based on the 3D location of the detected objects, which represent the relative 3D location of all the tracked objects. Figure 5 shows an example of the depth map, where Figure 5(a) shows the result of detect objects and Figure 5(b) shows the corresponding depth map. The depth map depicts the relative distance between the detected object and the camera. The higher intensity (brighter) means that the detected object is closer to the camera. By using the depth map, we can roughly assess whether an object is occluded by other objects based on the visibility $v_i\in[0,1]$:

$$v_i=\frac{\text{visible area of the } i^{\text{th}} \text{ target}}{\text{total area of the } i^{\text{th}} \text{ target}}, \tag{9}$$

and if $v_i=1$, it means the i^{th} target is totally visible; if $0<v_i<1$, it implies the i^{th} target is partially occluded; otherwise, it is fully occluded by other targets. As shown in Figure 5(a), all of the five objects are totally visible. So, the visibility should be set to $v_i=1$.

4.2. CMK Tracking. In traditional kernel-based tracking, a histogram including spatial and color information is usually used to represent the target and candidate model. During the histogram extraction, the contribution of a pixel is determined by the distance between the pixel and the kernel center. In [43], the tracking problem for maximizing the similarity $\text{simi}(\mathbf{x})$ is formulated as locating $\mathbf{x}$ that maximizes the probability density function (pdf) $f(\mathbf{x})$:



(a) (b)

Figure 5: Example of the depth map, showing (a) tracked objects and (b) the relative depth map.

$$f(\mathbf{x}) = \frac{\sum_{i=0}^{N_h} \omega_i k\left(\left\|(\mathbf{x} - \mathbf{z_i})/h\right\|^2\right)}{\sum_{i=0}^{N_h} k\left(\left\|(\mathbf{x} - \mathbf{z_i})/h\right\|^2\right)}, \tag{10}$$

where $\mathbf{x}$ is the kernel center; the subscript $\mathbf{i}$ represents each pixel location inside the kernel; $k(\cdot)$is a kernel function with a convex and monotonic decreasing kernel profile. $\mathbf{z_i}$ and ω_i are the position to be considered and the weight of a pixel, respectively; h is the bandwidth of the kernel.

After back-projecting the 2D locations to 3D locations of the detected object in the pose estimation stage, we use the depth CMK tracking technique to track them. The objective of depth CMK tracking is to find the candidate model that has the highest similarity to the target model, which is composed of multiple kernels with prespecified constraints in 3D space. For an object described by N_k kernels, the total cost function $J(X)$ is defined as the sum of N_k individual kernel cost functions $J_k(X)$, which is inversely proportional to the similarity:

$$J(X) = \sum_{k=1}^{N_k} J_k(X), \quad \frac{J_k(X) \propto 1}{\text{simi}_k(X)}, \tag{11}$$

where $\text{simi}_k(X)$ is the similarity function at the location $X \in R^3$. In addition, the constraint function $C(X)$ is used to confine the kernels according to their spatial interrelationships, and in order to maintain the relative location of each kernel, the constraint function needs to be set by $C(X) = 0$. Thus, the problem is further formulated as

$$\widehat{X} = \text{argmin}_X J(X), \quad \text{subject to } C(X) = 0. \tag{12}$$

However, when the object is occluded by other objects, not all the kernels in the object can be used for matching. To overcome this issue, we assigned an adaptively adjustable weight w_k to each kernel within the object. So, the cost function for the i^{th} target is as follows:

$$J^i(X) = \sum_{k=1}^{N_k} w_k^i \cdot J_k^i(X). \tag{13}$$

Taking the depth information into account, the visibility of each object can be set as a weight to handle global optimization. In other words, the total cost function in (11) becomes to

$$J(X) = \sum_{i=1}^{N_q} v_i \cdot J^i(X) = \sum_{i=1}^{N_q} v_i \cdot \left(\sum_{k=1}^{N_k} w_k^i \cdot J_k^i(X)\right), \tag{14}$$

where N_q is the number of the objects in the q^{th} image frame and w_k^i is a weight which is proportional to the similarity for the i^{th} target of each kernel N_k.

At the same time, the constraint functions $C(X) = 0$ must be considered to maintain the relative locations of the kernels. Figure 6(a) shows an example of the object was described by 2-kernel layouts in 2D space.

Unlike the work in [44] sets the constraints in 2D space, the constraints set in this paper are based on the 3D geometry. Without loss of generality, we discussed the 2-kernel case as shown in Figure 6(b), but it can be easily extended to the multikernel case. To represent an object in the 3D space, we define an object plane $(-n_q, \pi_q)$ for the object in the q^{th} image frame, where n_q is the normal vector of the q^{th} image frames, and π_q for the offset of the plane. In order to set the constraints properly, we start to estimate two auxiliary vectors, which are $u_q = -n_q \times g_q$ and $u_{1,2} = X_1 - X_2$. First, the distance between two kernel centers should be remained the same initial distance L, which implies

$$\left\|\mathbf{u}_{1,2}\right\|^2 = (L)^2. \tag{15}$$

Second, the angle ϕ_q between the vector u_q and $u_{1,2}$ and the angle ς_q between $-n_q$ and $u_{1,2}$ should be kept constant as well:

$$\begin{cases} \dfrac{u_q \cdot u_{1,2}}{\left\|u_q\right\|\left\|u_{1,2}\right\|} = \cos\left(\phi_q\right), \\ \dfrac{-n_q \cdot u_{1,2}}{\left\|-n_q\right\|\left\|u_{1,2}\right\|} = \cos\left(\varsigma_q\right). \end{cases} \tag{16}$$

These constraints can bind the kernels of the object to each other in the 3D space during the tracking. As shown in Figure 7(a), the constraint ϕ_q restricts the left-right movement of the kernels, and the constraint ς_q restricts the forward-backward movement of the kernels which is shown in Figure 7(b).

In order to gradually decrease the total cost function and maintain the constraints satisfied during the candidate model searching, the projected gradient method in [45] is

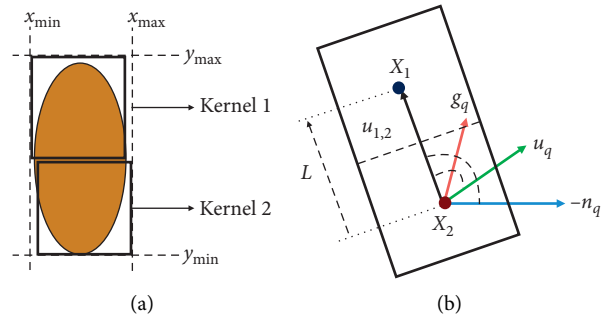


Figure 6: (a) Layout of an object with two kernels in 2D space. (b) Illustration of the 3D-based constraints in a 2-kernel layout.

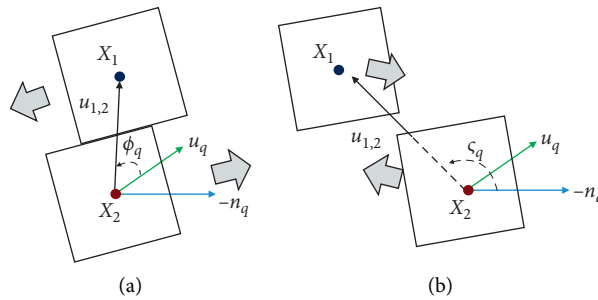


Figure 7: Constraints for binding two kernels in 3D space along the (a) left-right direction and the (b) forward-backward direction.

adopted to iteratively solve the constrained optimization problem. The basic concept of the method is to project the movement vector δ_X, i.e., the gradient vector of the $J(x)$, onto two orthogonal spaces. One is associated with decreasing the total cost function, and the other is responsible for satisfying the constraint function $C(X) = 0$:

$$\begin{aligned}\delta_X &= \alpha\left(-I + C_X\left(C_X^T C_X\right)^{-1} C_X^T\right) VWJ_X \\ &\quad + \left(-C_X\left(C_X^T C_X\right)^{-1} C_X\right) \\ &= \delta_X^A + \delta_X^B,\end{aligned} \tag{17}$$

where α is the size of searching step; I is a $3N_q \times 3N_q$ identity matrix,$\mathbf{C}(\mathbf{x}) = [c_1(\mathbf{x}), \ldots, c_m(\mathbf{x})]^T$ consists of m constraint functions, and $c_j(X)$: $\mathbb{R}^{3\cdot N_q \cdot N_k} \longrightarrow \mathbb{R}$ is the j^{th} constraint function; $\mathbf{V} = \begin{bmatrix} v_1 I_v & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & v_{N_q} I_v \end{bmatrix}$, where I_v is an $3N_k \times 3N_k$ identity matrix, which represents the visibility of kernels in the object; $\mathbf{W} = \begin{bmatrix} w_1 I_w & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & w_{N_k} I_w \end{bmatrix}$, where I_w is an $3N_q \times 3N_q$ identity matrix, which represents the similarity of the object.

As proved in [44], $\delta_{\mathbf{x}}^A$ and $\delta_{\mathbf{x}}^B$ have the following three characteristics. The first one is that $\delta_{\mathbf{x}}^A$ and $\delta_{\mathbf{x}}^B$ are orthogonal to each other. The second one is that moving along the $\delta_{\mathbf{x}}^A$ will decrease the total cost function $J(X)$ while keeping the same values of the constraint function $C(\mathbf{x})$. The last one is that moving along the $\delta_{\mathbf{x}}^B$ can lower the absolute values of constraint function $C(\mathbf{x})$. Owing to these three characteristics, the optimal solution can be reached in an iterative manner. The iteration is stopped until either the cost function and the absolute values of constraint are both lower than some given thresholds ε_j and ε_c, respectively, or the iteration count is larger than a threshold T (Algorithm 1 in [44]).

4.3. Hypothesized Association. Due to the occlusion or unreliable detection, objects may not be detected within a few frames. Therefore, some tracked targets cannot be successfully associated with the detections in subsequent frames. A hypothesized association which has been located by the CMK tracking with the best color similarity was inserted to consistently track a nonassociated target. By inserting hypothetical associations, it not only can improve the detection rate, but it also helps to continuously track the target. When an object is occluded, we can predict the 3D location by taking advantage of its 3D information, and a hypothesized association is thus used to pretend a possible detection. On the other hand, if a tracked target cannot be successfully associated to detection for several frames (empirically set as five frames in this work), then this target is considered as a missed target.

4.4. Kalman Filter Prediction and Update. Kalman filter is a traditional unscented transform-based state estimation method, which is used to approximate the mean and covariance of random variables after a nonlinear conversion. Most of tracking problems can be formulated as a state estimation problem. The tracking target can be regarded as a

Input:
Output: (g_k, φ_k)
(1) Initial frame number $N = 30$.
(2) Load a new frame f_k, k is the number of input frames.
(3) If $k < N$, set $D = [(g_1, \varphi_1)^T, \ldots, (g_k, \varphi_k)]$, go to step 6.
(4) If $\theta_{\text{rotation}} = |\theta_{k-1} - \theta_{k-N}| > \theta_{\text{threshold}}$, using $N^* = |1 - (2/\pi) \cdot \theta_{\text{rotation}}| \cdot N$ frames to estimate ground plane. Set $D = [(g_{k-N^*}, \varphi_{k-N^*})^T, \ldots, (g_{k-1}, \varphi_{k-1})^T]$ else set $D = [(g_{k-N}, \varphi_{k-N})^T, \ldots, (g_{k-1}, \varphi_{k-1})^T]$. Go to step 6.
(5) If the $\theta_{\text{rotation}} > (\pi/2)$ go to step 1.
(6) Input the D to RPCA and output the final (g_k, φ_k).

Algorithm 1: Adaptive ground plane estimation.

state, and the tracking problem is to predict and locate where the target (state) will appear in the next time. For this reason, the Kalman filter is widely used to solve tracking problems. The traditional Kalman filter is defined as follows:

$$\begin{aligned} \mathbf{x}_t &= \mathbf{F}_t \mathbf{x}_{t-1} + \mathbf{w}_{t-1}, \\ \mathbf{y}_t &= \mathbf{H}_t \mathbf{x}_t + \mathbf{v}_t, \end{aligned} \tag{18}$$

where $\mathbf{x}_t \in R^n$ and $\mathbf{y}_t \in R^m$ denote the state and measurement vector at the time step t, respectively; $\mathbf{F}_t$ is the state transition matrix; $\mathbf{H}_t$ is measurement matrix; $\mathbf{w}_{t-1} \sim N(0, Q)$ and $\mathbf{v}_t \sim N(0, R)$ are the system and measurement noise, and these two random variables are uncorrelated Gaussian white-noise sequence, with their covariance matrix Q and R, respectively.

In the stage of prediction, the predictions for state and error covariance are as follows:

$$\widehat{x}_t = \mathbf{F}_t \mathbf{x}_{t-1}, \tag{19}$$

$$\widehat{P}_t = \mathbf{F}_t \mathbf{P}_{t-1} \mathbf{F}_t^T + \mathbf{Q}_{t-1}. \tag{20}$$

After completing the measurement, the Kalman filter will be updated as follows:

$$\begin{aligned} \mathbf{K}_t &= \widehat{P}_t \mathbf{H}_t^T \left(\mathbf{H}_t \widehat{P}_t \mathbf{H}_t^T + \mathbf{R}_t \right)^{-1}, \\ \mathbf{x}_t &= \widehat{x}_t + \mathbf{K}_t \left(\mathbf{y}_t - \mathbf{H}_t \widehat{x}_t \right), \\ \mathbf{P}_t &= \left(\mathbf{I} - \mathbf{K}_t \mathbf{H}_t \right) \widehat{P}_t. \end{aligned} \tag{21}$$

The implementation of the Kalman filter algorithm is formulated as follows.

4.4.1. Initialization. For each object, the state vector is defined as $\mathbf{x}_t = [\, u_t \;\; v_t \;\; \dot{u}_t \;\; \dot{v}_t \;\; a_t \;\; b_t \,]^T$ and the measurement vector is defined as $\mathbf{y}_t = [\, u_t \;\; v_t \;\; a_t \;\; b_t \,]^T$, where (u_t, v_t), $(\dot{u}_t, \dot{v}_t)$, and (a_t, b_t) denote the object position, velocity, and size, respectively. Hence, the initial for the state transition matrix $\mathbf{F}_t$ and the measurement matrix $\mathbf{H}_t$ are defined as

$$\begin{aligned} \mathbf{F}_t &= \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \\ \mathbf{H}_t &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \end{aligned} \tag{22}$$

4.4.2. State Transition Matrix Update. In addition, the size of object in the image sequence will probably change when it is moving toward or away from the camera, and the extracted color histogram used for similarity measurement is highly dependent on the kernel size. On the other hand, when the multiple kernel tracking is performed, the result of segmentation might be no longer reliable for estimating the similarity due to occlusion. Hence, the state transition matrix needs to be modified adaptively to reflect the potential size changes. So, we embed the factor of kernel size into the matrix $\mathbf{F}_t$:

$$\mathbf{F}_t = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 + \dfrac{\beta \nabla f(h_x)}{a_{t-1}} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 + \dfrac{\beta \nabla f(h_y)}{b_{t-1}} \end{bmatrix}, \tag{23}$$

where β is the step size which also contains the smoothing factor; $\nabla f(h)$ is the derivative of the pdf with the kernel bandwidth h. Hence, the predict size of the object becomes to

$$\begin{bmatrix} \widehat{a}_t \\ \widehat{b}_t \end{bmatrix} = \begin{bmatrix} a_{t-1} + \beta \nabla f(h_x) \\ b_{t-1} + \beta \nabla f(h_y) \end{bmatrix}. \tag{24}$$

If the object is occluded so much that the average similarity value of all the kernels is lower than a certain threshold, the mechanism of state transition matrix update stops and $\mathbf{F}_t$ returns to the default setting as (22).

4.4.3. Measurement Noise Covariance Matrix Update. We use the object tracking result as a measurement to update the Kalman filter during the tracking. Although the system is robust under occlusion by using multiple kernels tracking, it still needs a mechanism to avoid the errors caused by incorrect measurements. It can be seen from (19) and (20) that not only does the Kalman gain $\mathbf{K}_t$ control the tradeoff between using the prediction and the measurement, but also it is inversely proportional to the measurement noise covariance matrix R. Hence, we can adaptively adjust the portion measurement contribution to avoid errors by changing the covariance matrix as follows:

$$\mathbf{R} = \begin{bmatrix} \sigma^2 \times J(X) & 0 & 0 & 0 \\ 0 & \sigma^2 \times J(X) & 0 & 0 \\ 0 & 0 & w^2 \times J(X) & 0 \\ 0 & 0 & 0 & h^2 \times J(X) \end{bmatrix}, \tag{25}$$

where $J(X)$ is the total cost function of all kernels; σ^2 is the predefined variance value, and w and h are the width and height of the kernel, respectively. With the help of the adaptively covariance matrix, if the total similarity between the candidate and the target is high, the diagonal term of the covariance matrix will be small. In this way, the Kalman gain will have a larger value, which will make the updated state closer to a reliable measurement.

5. Adaptive Ground Plane Estimation

Due to the unpredictability of driving road conditions, the ground plane estimated in the beginning may not be suitable for the entire image sequences. Therefore, the ground plane needs to be continuously reestimated based on the dynamic road conditions. In [33], the ground plane is reestimated and parameter smoothened every $f_g = 200$ frames to mitigate the adverse impact by the camera calibration noises. However, using a fixed number of frames for estimating the ground plane can affect the measurement accuracy when the camera is moving on a curve. In this paper, we propose to update the ground plane every single frame, based on an adaptively chosen N frames for parameter smoothing, by taking advantage of the camera rotation yaw angle calculated in the essential matrix calculation phase. The adaptive ground plane estimation algorithm is shown as follows.

In the algorithm, θ_k is the camera yaw angle at the k^{th} frame; (g_k, φ_k) is the ground plane at the k^{th} frame;$g_k \in R$ is the normal vector; and $\varphi_k \in R$ is the offset of the plane. D is a single $4 \times f_N$ matrix, and its elements are f_N ground planes, which is estimated by each pair of consecutive frames:

$$D = \left[\left(g_q, \varphi_q\right)^T, \ldots, \left(g_{q+f_N}, \varphi_{q+f_N}\right)^T \right]. \tag{26}$$

Due to the noisy camera calibrations and the unpredictability of road conditions, some ground planes (g_q, φ_q) may be unreliable; therefore, the robust principle component analysis (RPCA) [46] is applied to decompose a low-rank $4 \times f_N$ matrix A from D. The low-rank matrix's mean vector (g_k, φ_k) is considered to be our final ground plane, which is more robust to the noise contributed from the camera calibration and essential matrix calculation stage (see Section 3.2), derived from those f_N consecutive frames. Figure 8 shows an example of using a set of ground planes $\left\{ \left(g_q, \varphi_q\right)^T | q = 1, \ldots, f_N \right\}$ to estimate the final ground plane (g_k, φ_k). The gray planes are the image sequences converted from the driving recorder, and H is the camera height. The final ground plane for f_N consecutive frames (dot-line plane) is obtained from a set of ground planes (solid planes).

6. Experiment Results

In this section, we show experimental results of the proposed system on the Kitti datasets [47], which are taken with high quality dash cameras with motion pose ground truth and GPS information available. We test eight sequences (see Figure 9(a)), which are relatively short, and most of them are driving on a curvy road. Figure 9(b) shows the relative ground plane estimation results by applying our proposed method. We also test two of self-recorded video sequences captured around the University of Washington (UW) campus using a driving recorder mounted on a fixed height 1650 mm. And a more complex scenario in the ETHMS dataset, which includes multiple pedestrians on one scene, is also tested, and Table 1 shows the configurations of the tested videos.

6.1. The Relative Angular and Distance Errors. To demonstrate the accuracy of our proposed adaptive ground estimation, we compare the performance on the Kitti dataset with the following three different methods: the method in [4] is a stereo algorithm based on graphical model; the method in [17] formulates the ground plane estimation as a state continuous hidden Markov model where the hidden state contains ground plane; the method in [33] adopted the simultaneous localization and mapping (SLAM) technique to estimate the ground plane by using constant frames.

As in the method [17], the average relative angular error and distance error of the camera's motion are applied to evaluate the accuracy of the ground plane estimation. For the performance measurement, we calculate the camera poses and compare them with the given camera pose ground truth. The average relative angular and distance errors, which are

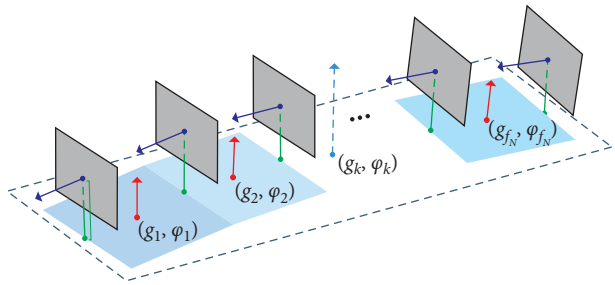
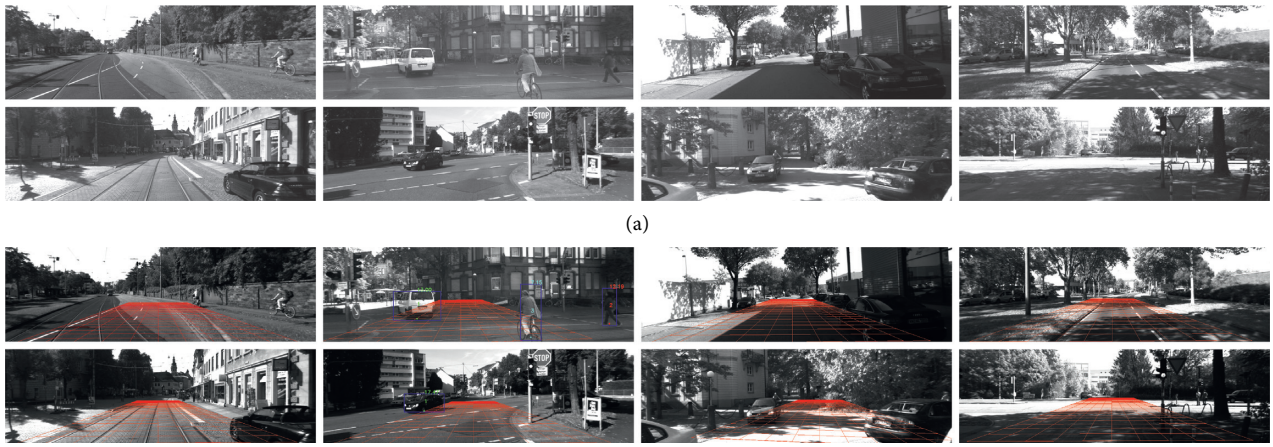


Figure 8: Example of the ground plane estimation.



(a)

(b)

Figure 9: Overview of the 8 sequences from the Kitti dataset and their relative ground plane estimation results: (a) the sequences 1–8 (row-wise starting top left), taken from the Kitti dataset; (b) the relative ground plane estimation result.

Table 1: Configurations of the datasets.

Sequence	Resolution	#Frames	Frame per second
Dataset seq#1	1242 × 375	77	15
Dataset seq#2	1242 × 375	155	15
Dataset seq#3	1242 × 375	447	15
Dataset seq#4	1242 × 375	233	15
Dataset seq#5	1242 × 375	154	15
Dataset seq#6	1242 × 375	384	15
Dataset seq#7	1242 × 375	87	15
Dataset seq#8	1242 × 375	106	15
UWcamp#1	1920 × 1080	1100	30
UWcamp#2	1920 × 1080	200	30
ETHMS #4	640 × 480	450	15

normalized by the path length, are given in Tables 2 and 3 separately.

Tables 2 and 3 show that the performance of our approach is better than the method [33] in both relative angular errors and comparable relative distance errors. That is because the estimated ground plane becomes more reliable after applying the adaptive ground plane estimation algorithm. Unlike the method in [33] that uses a constant number of frames to estimate the ground plane, our proposed method takes advantage of the estimated yaw angle in the camera pose to fight the adverse effects of the changing road conditions. Compared to the method used

Table 2: The average relative angular errors (DEG).

Dataset	Our method	Method [33]	Method [17]	Method [4]
1	0.06	0.11	0.02	0.8
2	0.05	0.20	0.07	1.22
3	0.03	0.08	0.04	0.27
4	0.01	0.03	0.23	0.92
5	0.06	0.06	0.01	0.41
6	0.03	0.08	0.07	0.39
7	0.05	0.10	0.20	3.06
8	0.10	0.59	0.11	1.68

in [17], our proposed scheme also shows better performance, except for the angular error in datasets 1 and 5, similarly except for the distance error in dataset 6 when compared with the method used in [4]. The major reason of the better performance is that our method can be well contributed by the noise reduction from the camera calibration and the unpredictability of road conditions as facilitated by taking advantage of the characteristics of adaptive-length RPCA.

6.2. Detection Performance. To demonstrate the detection performance of our proposed system, we compared it with three methods [33, 48, 49] with different human detectors on the ETHMS dataset, in terms of the detection rate and false

Table 3: The average relative distance errors (%).

Dataset	Our method	Method [33]	Method [17]	Method [4]
1	0.01	0.01	0.59	0.69
2	0.02	0.02	0.75	0.40
3	0.03	0.03	0.72	0.23
4	0.01	0.01	1.99	0.33
5	0.01	0.01	0.34	0.41
6	0.40	0.40	0.74	0.28
7	0.01	0.01	1.65	4.95
8	0.01	0.01	2.13	1.11

Table 4: Comparison of detection rate and FPPI.

Method	Detector	Detection rate (%)	FPPI
Method [48]	ISM	47	1.5
Method [48]	HOG	67.5	1
Method [49]	DPM	49.53	0.93
Method [49]	SP	51.86	0.92
Method [33]	DPM	75.58	0.89
Our method	DPM	75.71	0.82
Our method	YOLOv3	78.10	0.19

positive per image (FPPI). This shows the performance of inserting hypothesized association during tracking. The test results of the ETHMS dataset are shown in Table 4. The result shows that both the proposed method and the method in [33] are superior to the method in [48, 49]. Both methods further utilize the 3D information of the detected object, instead of only using 2D information in [48, 49]. They can effectively handle the occlusion issues. When compared with the method [33] with the DPM detector, the proposed method performs much better because it performs better in the tracking with the adaptive ground plane estimation, which results in increasing the detection rate and decreasing the FPPI. And compared with DPM and YOLOv3 detectors, the proposed method with YOLOv3 has a better performance due to the low false positive detection rate in the YOLOv3. Thanks to the proper insertion of hypothesized associations and the successive tracking, the detection rate of the proposed method can achieve about 78%. This implies that missing detection can be improved by the tracking techniques, and thus better detection results benefit the tracking performance.

6.3. Multiple Object Tracking Result. To demonstrate the tracking performance of our proposed system, we compare the performance with the following three different tracking methods: the method in [44] is a kernel-based human-tracking system which tracks a human in 2D space and without estimating the ground plane. The method in [50] uses the tracking-by-detection scheme to associate the detected objects by calculating their similarity. The method in [33] is a human tracking system which uses a constant number of frames to estimate the ground plane. To fairly evaluate the tracking performance for each method, we manually labeled 7302 locations as ground truth which includes 31 moving vehicles and 89 pedestrians across 3393 frames and also adopt the following metrics which are widely used in multiple object tracking (MOT) challenge [51].

(i) Multiple object tracking accuracy (MOTA): the measurement of tracking accuracy combines three sources of errors: false positive, false negative, and identity switches.

(ii) Multiple object tracking precision (MOTP): the measurement of object localization precision.

(iii) False positive (FP): the number of times of the system detects an object but the ground truth is not present in the image frame.

(iv) False negative (FN): the number of times of the system failed to detect an object but the ground truth is present in the image frame.

(v) ID switches (IDSs): the number of times two trajectories switch their IDs.

The comparison of the experimental results is shown in Table 5. The proposed method achieved the best performance in all of the metrics except for FN. The reason is that the CNN-based tracking by detection retains more foreground around the object regions. However, the extra extracted background information will also cause the increase in FP and IDS. The ability of the proposed depth CMK to deal with occlusion issues can be learned from the fact that there is less identity switching, while the other methods are tending to generate new object identities when occlusion occurs. To facilitate the comparison of experimental results, the red entries in Table 5 indicate that the best results in the corresponding columns and blue italics are the second best.

An additional typical example of performance comparison is shown in Figures 10 and 11, which both extract five continuous frames from 175 to 179 from the UW campus sequence 1. Figure 10 shows the tracking results in the method [33], which use a constant number to estimate ground plane. Figure 11 shows the tracking result in the proposed method, which takes advantage of the yaw angle from the camera pose to estimate the ground plane adaptively. From Figure 10, we can see that the camera mounted on the driving vehicle starts to change direction in the frame 175, and in the frame of 177, the distance of the vehicle to the camera sharply changed from 10.31 to 7.98, and then back to 8.31 in the frame 179. The estimated ground plane remains the same even when the vehicle starts to turn. Figure 11 shows the tracking performance of the proposed method using adaptive ground plane estimation, we can see that the distance of the vehicle gradually reduces from 10.51 to 8.44, and the ground plane keeps changing with the direction of the vehicle adaptively. It can be observed that the proposed method can track objects more continuously and effectively by using the adaptive ground plane estimation. Several object tracking results with estimated ground plane are shown in Figures 12–14, which show the tracking results on the UW campus sequence 2, Kitti datasets, and ETHMS datasets, respectively. The results show favorable performance of the proposed system, which not only can successively track objects but also estimate a reliable ground plane adaptively.

Table 5: The tracking performance between different methods.

Methods	MOTA (%)	MOTP (%)	FP	FN	IDS
Our method	79.7	95.8	143	1313	29
Method [33]	76.2	92.1	268	1416	53
Method [44]	63.9	82.6	590	1955	91
Method [50]	7.8	90.7	316	1223	82

(a) (b) (c) (d) (e)

Figure 10: Tracking results in the method [33] without adaptive ground plane estimation on UW campus sequence #1. (a) Frame 175. (b) Frame 176. (c) Frame 177. (d) Frame 178. (e) Frame 179.

(a) (b) (c) (d) (e)

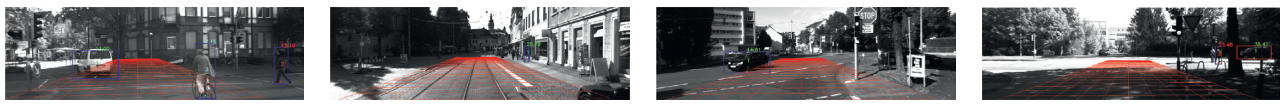
Figure 11: Tracking result in our method with adaptive ground plane estimation on UW campus sequence #1. (a) Frame 175. (b) Frame 176. (c) Frame 177. (d) Frame 178. (e) Frame 179.

(a) (b) (c) (d)

Figure 12: Tracking results with the estimated ground plane on UW campus sequence #2. (a) Frame 3. (b) Frame 4. (c) Frame 5. (d) Frame 6.

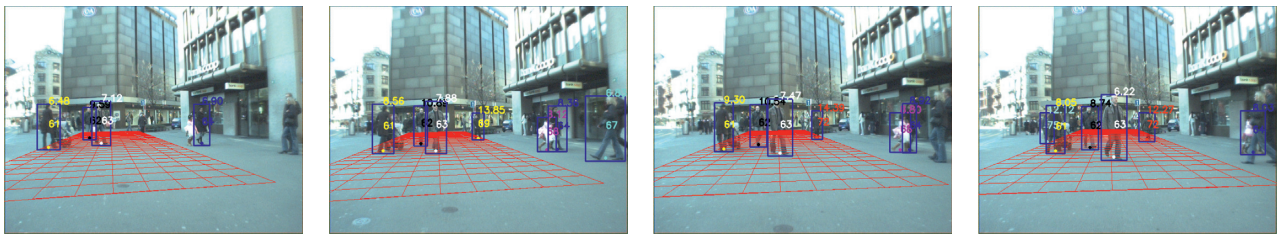
6.4. Runtime Performance. Apart from the detectors, all the experiments are processed on a laptop with an Intel Core i7, 2.2 GHz CPU with 8GB DDR. The implementation is constructed by C/C++, and the experimental settings are described as follows: in the structure from motion phase, the proposed system uses the Harris corner detector to extract 1000 features initially, which are tracked by a KLT tracker. And these corresponding feature points are used to estimate the camera pose. In object detection, the pretrained YOLOv3 detectors are independently used in the proposed system to detect objects such as human and vehicle. In the depth CMK tracking, a depth map is constructed to describe the relative 3D locations of all the tracked objects firstly, and the histogram of objects is constructed based on the HSV color space with a roof kernel; then, the K-L distance is used for all the similarity-related measurements. Table 6 shows the running time of the proposed system on different datasets with different image resolutions.

6.5. Discussion. In this paper, we proposed an adaptive ground plane estimation algorithm-based tracking system. Existing ground plane estimation methods are required to meet significant assumptions, such as the ground plane is the largest plane in the scene and the ground plane is constant in color or texture. These assumptions are not practical in cluttered or dynamic environments, especially not suitable for driving environments. Our method can robustly estimate



(a) (b) (c) (d)

Figure 13: Tracking results with the estimated ground plane on the Kitti datasets. (a) Kitti dataset 2. (b) Kitti dataset 5. (c) Kitti dataset 6. (d) Kitti dataset 8.



(a) (b) (c) (d)

Figure 14: Tracking results with the estimated ground plane on the ETHMS datasets. (a) Frame 175. (b) Frame 185. (c) Frame 191. (d) Frame 196.

Table 6: Runtime on different image resolutions.

Dataset	Resolution	Average runtime (fps)
Kitti	1242 × 375	0.87
UW campus	1920 × 1080	0.65
ETHMS	640 times 480	0.95

the ground plane on a moving camera with nonrestrictive assumption: the camera is mounted on a fixed height of the vehicle.

Combining the adaptive ground plane estimation, object detection, Kalman filter framework, and efficient depth CMK tracking techniques, the proposed tracking system can not only track the object effectively but also robustly handle occlusion during tracking. Nevertheless, several limitations are still existed. First, the proposed approach adopts the tracking-by-detection scheme to detect and then track objects, and this implies that the method highly relies on the detection results. However, if the quality of video sequences is not sufficient for the object detectors, the proposed tracking system is not able to perform well on the poor detection results. More specifically, the positive detection of a target can always trigger the tracking of a specific object. In other words, the proposed method may not work well at night or some cases of insufficient lighting. Second, the proposed method effectively estimates ground planes based on certain video frames when the vehicle moves on flat roads, but if the roads are severely bumpy, it will produce less reliable estimation, resulting in larger error of the object back-projection and impacting accuracy of the reprojected 3D information. Hence, the proposed method is not reliable for the unmanned aerial vehicle, because its height dynamically changes and then infers unreliable 3D information of objects.

In the future, we will focus on improving the performance of the algorithm by enhancing the accuracy of the object detection algorithms. In addition, we will also test our algorithms on video sequences that have higher outdoor complexity and more objects visible in the scene.

7. Conclusion

We propose a robust object tracking system and ground plane estimation simultaneously in a dashcam mounted on a free-moving vehicle. The proposed system effectively integrates the object detection, ground plane estimation, CMK tracking, and Kalman filter framework to relocate the objects in 3D space, and the estimated camera yaw angle has been adopted into the adaptive ground plane estimation. With the depth CMK tracking, the 3D positions of the detected targets are updated on the more reliable ground plane and occlusion issue is also handled in the tracking system. The experimental result shows that the proposed method greatly improved the tracking performance. Such tracking system can be regarded as a key component for high-level applications, such as video analysis in a large scale of the mobile network. Besides, the proposed framework can also be futher applied to the advanced driver assistance system (ADAS).

Data Availability

The Kitti dataset used to support the findings of this study may be released upon application to the KITTI Vision Benchmark Suite, which is a project of Karlsruhe Institute of Technology and Toyota Technological Institute at Chicago. The dataset can be downloaded for free at this web page http://www.cvlibs.net/datasets/kitti/raw_data.php. The ETHMS dataset can be downloaded on the following web page https://data.vision.ee.ethz.ch/cvl/aess/dataset/#pami09. Requests for self-recorded UW data, 6/12 months, after the publication of this article, will be considered by the corresponding author.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the 111 Project, under grant B17007, and in part by the China Scholarship Council Funding.

References

[1] T. Liu, Y. Liu, Z. Tang, and J.-N. Hwang, "Adaptive ground plane estimation for moving camera-based 3d object tracking," in *Proceedings of the 2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP)*, pp. 1–6, IEEE, London, UK, October 2017.

[2] L. Ladickỳ, P. Sturgess, C. Russell et al., "Joint optimization for object class segmentation and dense stereo reconstruction," *International Journal of Computer Vision*, vol. 100, no. 2, pp. 122–133, 2012.

[3] D. Maier and M. Bennewitz, "Appearance-based traversability classification in monocular images using iterative ground plane estimation," in *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4360–4366, IEEE, Vilamoura, Portugal, October 2012.

[4] A. Geiger, J. Ziegler, and C. Stiller, "Stereoscan: Dense 3D reconstruction in real-time," in *Proceedings of the 2011 IEEE Intelligent Vehicles Symposium (IV)*, pp. 963–968, IEEE, Baden-Baden, Germany, June 2011.

[5] C. Yuan and G. Medioni, "3D reconstruction of background and objects moving on ground plane viewed from a moving camera," in *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, pp. 2261–2268, IEEE, New York, NY, USA, June 2006.

[6] A. Cherian, V. Morellas, and N. Papanikolopoulos, "Accurate 3D ground plane estimation from a single image,," in *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*, pp. 2243–2249, IEEE, Kobe, Japan, May 2009.

[7] J. Arróspide, L. Salgado, M. Nieto, and R. Mohedano, "Homography-based ground plane detection using a single on-board camera," *IET Intelligent Transport Systems*, vol. 4, no. 2, pp. 149–160, 2010.

[8] D. Conrad and G. N. DeSouza, "Homography-based ground plane detection for mobile robot navigation using a modified em algorithm," in *Proceedings of the 2010 IEEE International Conference on Robotics and Automation*, pp. 910–915, IEEE, Anchorage, AK, USA, May 2010.

[9] P. Ke, C. Meng, J. Li, and Y. Liu, "Homography-based ground area detection for indoor mobile robot using binocular cameras," in *Proceedings of the 2011 IEEE 5th International Conference on Robotics, Automation and Mechatronics (RAM)*, pp. 30–34, IEEE, Qingdao, China, September 2011.

[10] S. Kumar, A. Dewan, and K. M. Krishna, "A bayes filter based adaptive floor segmentation with homography and appearance cues," in *Proceedings of the Eighth Indian Conference on Computer Vision, Graphics and Image Processing*, pp. 1–8, Mumbai, India, December 2012.

[11] R. Labayrade, D. Aubert, and J.-P. Tarel, "Real time obstacle detection in stereovision on non flat road geometry through "v-disparity" representation," in *Proceedings of the Intelligent Vehicle Symposium, 2002*, pp. 646–651, IEEE, Yangzhou, China, June 2002.

[12] Y. Lang, H. Wu, T. Amano, and Q. Chen, "An iterative convergence algorithm for single/multi ground plane detection and angle estimation with rgb-d camera," in *Proceedings of the 2015 IEEE International Conference on Image Processing (ICIP)*, pp. 2895–2899, IEEE, Québec City, Canada, September 2015.

[13] J. Zhao, J. Katupitiya, and J. Ward, "Global correlation based ground plane estimation using v-disparity image," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pp. 529–534, IEEE, Roma, Italy, April 2007.

[14] Z. Jin, T. Tillo, and F. Cheng, "Depth-map driven planar surfaces detection," in *Proceedings of the 2014 IEEE Visual Communications and Image Processing Conference*, pp. 514–517, IEEE, Valletta, Malta, December 2014.

[15] D. Kırcalı and F. B. Tek, "Ground plane detection using an rgb-d sensor," in *Proceedings of the Information Sciences and Systems 2014*, pp. 69–77, Springer, Shenzhen, China, April 2014.

[16] P. Skulimowski, M. Owczarek, and P. Strumiłło, "Ground plane detection in 3D scenes for an arbitrary camera roll rotation through "v-disparity" representation," in *Proceeding of the 2017 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pp. 669–674, IEEE, Prague, Czech Republic, September 2017.

[17] R. Dragon and L. V. Gool, "Ground plane estimation using a hidden markov model," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4026–4033, Columbus, OH, USA, June 2014.

[18] R. Dragon, B. Rosenhahn, and J. Ostermann, "Multi-scale clustering of frame-to-frame correspondences for motion segmentation," in *Proceedings of the European Conference on Computer Vision*, pp. 445–458, Springer, Florence, Italy, October 2012.

[19] Y. Man, X. Weng, X. Li, and K. Kitani, "Groundnet: monocular ground plane normal estimation with geometric consistency," in *Proceedings of the 27th ACM International Conference on Multimedia*, pp. 2170–2178, Nice, France, October 2019.

[20] M. W. McDaniel, T. Nishihata, C. A. Brooks, and K. Iagnemma, "Ground plane identification using lidar in forested environments," in *Proceedings of the 2010 IEEE International Conference on Robotics and Automation*, pp. 3831–3836, IEEE, Anchorage, AK, USA, May 2010.

[21] F. Mufti, R. Mahony, and J. Heinzmann, "Robust estimation of planar surfaces using spatio-temporal ransac for applications in autonomous vehicle navigation," *Robotics and Autonomous Systems*, vol. 60, no. 1, pp. 16–28, 2012.

[22] D. Borrmann, J. Elseberg, K. Lingemann, and A. Nüchter, "The 3d hough transform for plane detection in point clouds: a review and a new accumulator design," *3D Research*, vol. 2, no. 2, p. 3, 2011.

[23] D. Scaramuzza, F. Fraundorfer, and R. Siegwart, "Real-time monocular visual odometry for on-road vehicles with 1-point ransac," in *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*, pp. 4293–4299, IEEE, Kobe, Japan, May 2009.

[24] X. Qian and C. Ye, "Ncc-ransac: A fast plane extraction method for 3-D range data segmentation," *IEEE Transactions on Cybernetics*, vol. 44, no. 12, pp. 2771–2783, 2014.

[25] S. Se and M. Brady, "Ground plane estimation, error analysis and applications," *Robotics and Autonomous Systems*, vol. 39, no. 2, pp. 59–71, 2002.

[26] J.-P. Tardif, Y. Pavlidis, and K. Daniilidis, "Monocular visual odometry in urban environments using an omnidirectional camera," in *Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2531–2538, IEEE, Nice, France, September 2008.

[27] B. Micusik and J. Kosecka, "Piecewise planar city 3d modeling from street view panoramic sequences," in *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2906–2912, IEEE, Miami Beach, FL, USA, June 2009.

[28] C. Zhang and S. Czarnuch, "Perspective independent ground plane estimation by 2D and 3D data analysis," *IEEE Access*, vol. 8, pp. 82024–82034, 2020.

[29] D. Holz, S. Holzer, R. B. Rusu, and S. Behnke, "Real-time plane segmentation using rgb-d cameras," in *Robot Soccer World Cup*, pp. 306–317, Springer, Berlin, Germany, 2011.

[30] S. Choi, J. Park, J. Byun, and W. Yu, "Robust ground plane detection from 3D point clouds," in *Proceedings of the 2014 14th International Conference on Control, Automation and Systems (ICCAS 2014)*, pp. 1076–1081, IEEE, Seoul, Korea, October 2014.

[31] T. Liu and Y. Liu, "Deformable model-based vehicle tracking and recognition using 3-D constrained multiple-Kernels and Kalman filter," *IEEE Access*, vol. 9, 2021.

[32] J. Heikkila and O. Silven, "A four-step camera calibration procedure with implicit image correction," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1106–1112, IEEE, San Juan, Puerto Rico, June 1997.

[33] K.-H. Lee, J.-N. Hwang, G. Okopal, and J. Pitton, "Ground-moving-platform-based human tracking using visual slam and constrained multiple kernels," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 12, pp. 3602–3612, 2016.

[34] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[35] B. Liang, N. Pears, and Z. Chen, "Affine height landscapes for monocular mobile robot obstacle avoidance," *In Proceedings of Intelligent Autonomous Systems*, vol. 8, pp. 863–872, 2004.

[36] J. Zhou and B. Li, "Homography-based ground detection for a mobile robot platform using a single camera," in *Proceedings 2006 IEEE International Conference on Robotics and Automation*, pp. 4100–4105, Beijing, China, June 2006.

[37] J. Zhou and B. Li, "Robust ground plane detection with normalized homography in monocular sequences from a robot platform," in *Proceedings of the 2006 International Conference on Image Processing*, pp. 3017–3020, Varzim, Portugal, September 2006.

[38] N. Simond and M. Parent, "Obstacle detection from ipm and super-homography," in *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4283–4288, San Diego, CA, USA, November 2007.

[39] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, June 2016.

[40] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, pp. 886–893, IEEE, San Diego, CA, USA, June 2005.

[41] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2009.

[42] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," 2018.

[43] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564–577, 2003.

[44] C.-T. Chu, J.-N. Hwang, H.-I. Pai, and K.-M. Lan, "Tracking human under occlusion based on adaptive multiple kernels with projected gradients," *IEEE Transactions on Multimedia*, vol. 15, no. 7, pp. 1602–1615, 2013.

[45] J. A. Snyman, *Practical Mathematical Optimization*, Springer, Berlin, Germany, 2005.

[46] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma, "Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization," in *Advances in Neural information Processing Systems*, pp. 2080–2088, Springer, Berlin, Germany, 2009.

[47] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3354–3361, IEEE, Providence, Rhode Island, June 2012.

[48] A. Ess, B. Leibe, K. Schindler, and L. Van Gool, "Robust multiperson tracking from a mobile platform," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 10, pp. 1831–1846, 2009.

[49] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes, "Globally-optimal greedy algorithms for tracking a variable number of objects," in *Proceedings of the CVPR 2011*, pp. 1201–1208, IEEE, Colorado Spring, CO, USA, June 2011.

[50] A. Milan, S. H. Rezatofighi, A. Dick, I. Reid, and K. Schindler, "Online multi-target tracking using recurrent neural networks," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, 2017.

[51] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, "Mot16: a benchmark for multi-object tracking," 2016, http://arxiv.org/abs/1603.00831.