




Article

# State Transition for Statistical SLAM Using Planar Features in 3D Point Clouds

Amirali Khodadadian Gostar <sup>1</sup>, Chunyun Fu <sup>2,\*</sup>, Weiqin Chuah <sup>1</sup>,  
Mohammed Imran Hossain <sup>1</sup>, Ruwan Tennakoon <sup>1</sup>, Alireza Bab-Hadiashar <sup>1</sup> and  
Reza Hoseinnezhad <sup>1</sup>

<sup>1</sup> School of Engineering, RMIT University, Melbourne VIC 3001, Australia; amirali.khodadadian@rmit.edu.au (A.K.G.); s3538114@student.rmit.edu.au (W.C.); s3638951@student.rmit.edu.au (M.I.H.); ruwan.tennakoon@rmit.edu.au (R.T.); abh@rmit.edu.au (A.B.-H.); rezah@rmit.edu.au (R.H.)

<sup>2</sup> State Key Laboratory of Mechanical Transmissions, School of Automotive Engineering, Chongqing University, Chongqing 400044, China

\* Correspondence: fuchunyun@cqu.edu.cn; Tel.: +86-23-6510-6243

Received: 11 February 2019; Accepted: 1 April 2019; Published: 3 April 2019



**Abstract:** There is a large body of literature on solving the SLAM problem for various autonomous vehicle applications. A substantial part of the solutions is formulated based on using statistical (mainly Bayesian) filters such as Kalman filter and its extended version. In such solutions, the measurements are commonly some point features or detections collected by the sensor(s) on board the autonomous vehicle. With the increasing utilization of scanners with common autonomous cars, and availability of 3D point clouds in real-time and at fast rates, it is now possible to use more sophisticated features extracted from the point clouds for filtering. This paper presents the idea of using planar features with multi-object Bayesian filters for SLAM. With Bayesian filters, the first step is prediction, where the object states are propagated to the next time based on a stochastic transition model. We first present how such a transition model can be developed, and then propose a solution for state prediction. In the simulation studies, using a dataset of measurements acquired from real vehicle sensors, we apply the proposed model to predict the next planar features and vehicle states. The results show reasonable accuracy and efficiency for statistical filtering-based SLAM applications.

**Keywords:** simultaneous localization and mapping; planar features; plane parameters; transition model; Bayesian filters

## 1. Introduction

Simultaneous Localization and Mapping (SLAM) is often considered as one of the main challenges in the field of robotics and autonomous vehicles [1,2]. The aim of SLAM is to build a map of an unknown environment while simultaneously determining the location of the vehicle within this map. Neither the map nor the vehicle position are known in advance. However, a kinematic model of the vehicle motion is assumed to be known a priori, and the unknown environment is populated with artificial or natural landmarks.

There is a large body of literature on solving the SLAM problem for various autonomous vehicle applications. Many of the solutions are formulated based on using a Bayesian filter that recursively propagates the distribution of the vehicle's dynamic states (and sometimes the map features). Davison [3] proposed a solution based on using Extended Kalman Filters (EKF) to track the location of each map feature which was extracted from the sensor measurements. To improve the accuracy of this method, Thrun et al. [4] proposed the Sparse Extended Information Filter.

The idea is to optimize the calculations required by the EKF, via taking advantage of sparse matrices. Montemerlo et al. [5] proposed a Rao-Blackwellised particle filter-based method called FastSLAM. In this method, the odometry uncertainties are rectified by adding random offsets to the odometry data for each particle.

In such solutions, to simultaneously estimate both vehicle and landmark locations, the Bayesian filter needs to employ two models: an observation model, and a motion model. With regards to the observations, the vehicle must be equipped with a sensor or set of sensors that produce measurements related to the surrounding landmark locations. The most common examples are LIDAR [6–8], RGB camera [9–12], RGB-D camera [13,14] and sonar [15] sensors. The sensory measurements usually have range limitations, and contain measurement noise. The raw measurements are normally processed further to extract point features. An on-board sensory system on the vehicle is usually in place to collect the features in real-time.

With the advancement of sensor technology and decreasing cost of LIDAR scanners, the generation of an adequate 3D point cloud is now possible. A LIDAR works by sending a laser pulse towards an object and measuring the time it takes to reflect from the object and return to the sensor. Because of its high accuracy and ease of use, a LIDAR sensor has become a common choice for SLAM purposes. Having the generated point cloud, it is possible to extract features that are more complex (and convey more information) than point features. Employing more information-rich features as observations in a Bayesian filter within a SLAM algorithm can increase the algorithm efficiency and reduce the computational cost. For example employing planar features instead of 3D points could considerably enhance the algorithm efficiency, as the number of planes is significantly smaller than the number of points in a typical 3D point cloud. However, in order to properly use these features in the Bayesian framework, an appropriate motion model needs to be devised.

In human-inhabited environments, buildings and large facilities are currently almost ubiquitous and their geometric profiles comprise a large number of plane shapes. Out of all these shapes, the major ones naturally create “obstacles” or define “edges” beyond which the vehicles cannot protrude, and these planes can be extracted and employed as features or landmarks in SLAM applications. Indeed, due to the geometric simplicity of the plane shapes and their abundance in human-inhabited environments, planar features have attracted increasing attention from both the computer graphics and robotics community in recent years [16–18]. As for the smaller plane shapes, they may be from the profiles of small objects or even moving objects such as vehicles. These small planes can be readily excluded from the set of landmarks by applying a threshold to the dimension of planes.

This paper presents for the first time the idea of using planar features extracted from a 3D point cloud, instead of point features, for Bayesian SLAM filters. The major contribution of this study lies in proposing a novel transition model that predicts the parameters of planar features extracted from a 3D point cloud. With Bayesian filters, the first step is prediction, where the object state densities are propagated to the next time step based on a stochastic transition model (motion model). We present how such a transition model can be developed, and propose a solution for state prediction. In the simulation studies, using a dataset of measurements acquired from real vehicle sensors, we apply the model to predict the planar features at the next time. The results show reasonable accuracy and efficiency for statistical filtering-based SLAM applications.

The proposed transition model consists of two sub-models: the plane transition model and the vehicle transition model. The latter one is associated with localization; namely, it predicts the vehicle states at time  $k + 1$  using the state information at time  $k$ . The former one is associated with mapping, which predicts the feature parameters at time  $k + 1$  based on the feature information at time  $k$  as well as the predicted vehicle states resulting from the latter model. The accuracy and effectiveness of the proposed transition model are verified using real-world point cloud measurements from the KITTI dataset [19]. The graphical and numerical simulation results show that the predicted planar features resulting from the proposed transition model closely match the measured features (i.e., segmented 3D

planes) at time  $k + 1$ , in terms of the plane dimensions (plane area and vertex distance) and the plane orientation (normal vector).

The rest of the paper is organized as follows. Section 2 provides relevant theoretical background. Section 3 introduces the details of the proposed transition model which comprises two sub-models: the plane transition model and the vehicle transition model. Section 4 explains the detailed procedure for implementation of the proposed transition model. Section 5 provides verification of the proposed transition model by means of graphical and numerical simulation results. Section 6 concludes the paper.

## 2. Background

The proposed transition model is mainly designed to be used with a recently developed Random Finite Set (RFS) based Bayesian filter—the Labeled Multi-Bernoulli (LMB) filter [20]. In this section, we provide the theoretical background necessary for understanding this filter, followed by a brief introduction to the KITTI dataset [19] which was used for simulation studies in this work.

### 2.1. Labeled Multi-Bernoulli Filter

In this section, we introduce the Labeled Multi-Bernoulli (LMB) filter [20]. We adopt the same notation used in [20] where the single-object states are denoted by lower-case letters, e.g.,  $x$ ,  $\mathbf{x}$  and multi-object states by upper-case letters, e.g.,  $X$ ,  $\mathbf{X}$ . In order to distinguish between labeled and unlabeled states and their distributions, the labeled one is shown by bolded letters, e.g.,  $\mathbf{x}$ ,  $\mathbf{X}$ , etc., spaces by blackboard bold, e.g.,  $\mathbb{X}$ ,  $\mathbb{L}$ ,  $\mathbb{C}$ , etc., and the class of finite subsets of a space  $\mathbb{X}$  by  $\mathcal{F}(\mathbb{X})$ . Following [20], throughout the paper, the standard inner product notation is used and denoted by  $\langle f, g \rangle \triangleq \int f(x)g(x)dx$ , the generalized Kronecker delta is denoted by

$$\delta_Y(X) \triangleq \begin{cases} 1, & \text{if } X = Y \\ 0, & \text{otherwise} \end{cases} ,$$

and the inclusion function, a generalization of the indicator function, by

$$1_Y(X) \triangleq \begin{cases} 1, & \text{if } X \subseteq Y \\ 0, & \text{otherwise} \end{cases} .$$

The LMB RFS is completely described by its components  $\pi = \{(r^{(\ell)}, p^{(\ell)}) : \ell \in \mathbb{L}\}$ . The LMB RFS density is given by  $\pi(\mathbf{X}) = \Delta(\mathbf{X})w(\mathcal{L}(\mathbf{X})) [p]^{\mathbf{X}}$ , where  $p(x, \ell) = p^{(\ell)}(x)$  and  $w(L) = \prod_{i \in \mathbb{L}} (1 - r^{(i)}) \prod_{\ell \in L} \frac{1_{\mathbb{L}}(\ell)r^{(\ell)}}{(1-r^{(\ell)})}$  comprising a single component [20]. The LMB multi-target Bayes recursion propagates multi-target posterior density at each time step according to the Chapman–Kolmogorov and the Bayes rule.

#### 2.1.1. Prediction

Assume that the prior and birth labeled multi-Bernoulli sets are modeled as follows:

$$\pi(\mathbf{X}) = \Delta(\mathbf{X})w(\mathcal{L}(\mathbf{X})) [p]^{\mathbf{X}} \quad (1)$$

$$\pi_B(\mathbf{X}) = \Delta(\mathbf{X})w_B(\mathcal{L}(\mathbf{X})) [p_B]^{\mathbf{X}} \quad (2)$$

where

$$w(L) = \prod_{i \in \mathbb{L}} (1 - r^{(i)}) \prod_{\ell \in L} \frac{1_{\mathbb{L}}(\ell)r^{(\ell)}}{1 - r^{(\ell)}} , \quad (3)$$

$$w_B(L) = \prod_{i \in \mathbb{B}} (1 - r_B^{(i)}) \prod_{\ell \in L} \frac{1_{\mathbb{B}}(\ell)r_B^{(\ell)}}{1 - r_B^{(\ell)}} , \quad (4)$$

$$p(x, \ell) = p^{(\ell)}(x) \quad (5)$$

$$p_B(x, \ell) = p_B^{(\ell)}(x). \quad (6)$$

with state space  $\mathbb{X}$  and label space  $\mathbb{L}_+ = \mathbb{B} \cup \mathbb{L}$  and with the condition  $\mathbb{B} \cap \mathbb{L} = \emptyset$ . The predicted multi-object distribution is then a LMB RFS and given by

$$\pi_+(\mathbf{X}) = \Delta(\mathbf{X})w_+(\mathcal{L}(\mathbf{X})) [p_+]^{\mathbf{X}} \quad (7)$$

where

$$w_+(I_+) = w_S(I_+ \cap \mathbb{L})w_B(I_+ \cap \mathbb{B}) \quad (8)$$

$$w_S(L) = \left(1 - r^{(\cdot)}\eta_S(\cdot)\right)^{\mathbb{L}} \left(\frac{r^{(\cdot)}\eta_S(\cdot)}{1 - r^{(\cdot)}\eta_S(\cdot)}\right)^L, \quad (9)$$

$$\eta_S(\ell) = \langle p_S(\cdot, \ell), p(\cdot, \ell) \rangle \quad (10)$$

$$p_+(x, \ell) = 1_{\mathbb{L}}(\ell)p_{+,S}(x, \ell) + 1_{\mathbb{B}}(\ell)p_B(x, \ell) \quad (11)$$

$$p_{+,S}(x, \ell) = \frac{\langle p_S(\cdot, \ell)f(x|\cdot, \ell), p(\cdot, \ell) \rangle}{\eta_S(\ell)} \quad (12)$$

where  $p_S(\cdot|\ell)$  is the survival probability of an object and  $f(x|\cdot, \ell)$  is the single-object transition model. Thus, if the multi-target posterior density is an LMB RFS with parameter set  $\pi = \{(r^{(\ell)}, p^{(\ell)}) : \ell \in \mathbb{L}\}$  with state space  $\mathbb{X}$  and label space  $\mathbb{L}$  and the birth model is also an LMB RFS with parameter set  $\pi_B = \{(r_B^{(\ell)}, p_B^{(\ell)}) : \ell \in \mathbb{B}\}$  with state space  $\mathbb{X}$  and label space  $\mathbb{B}$  then the predicted multi-target density is also an LMB RFS with state space  $\mathbb{X}$  and label space  $\mathbb{L}_+ = \mathbb{B} \cup \mathbb{L} (\mathbb{B} \cap \mathbb{L} = \emptyset)$  and it is given by

$$\pi_+ = \{(r_{+,S}^{(\ell)}, p_{+,S}^{(\ell)}) : \ell \in \mathbb{L}\} \cup \{(r_B^{(\ell)}, p_B^{(\ell)}) : \ell \in \mathbb{B}\} \quad (13)$$

where

$$r_{+,S}^{(\ell)} = \eta_S(\ell)r^{(\ell)}, \quad (14)$$

$$p_{+,S}^{(\ell)} = \frac{\langle p_S(\cdot, \ell)f(x|\cdot, \ell), p(\cdot, \ell) \rangle}{\eta_S(\ell)}, \quad (15)$$

for more details see— [20]—proposition 2.

### 2.1.2. Update

If the predicted multi-target density is an LMB RFS with parameter set  $\pi_+ = \{(r_+^{(\ell)}, p_+^{(\ell)}) : \ell \in \mathbb{L}_+\}$ , the multi-target posterior is then given by

$$\pi(\cdot|Z) = \{(r^{(\ell)}, p^{(\ell)}(\cdot)) : \ell \in \mathbb{L}_+\} \quad (16)$$

where

$$r^{(\ell)} = \sum_{(I_+, \theta) \in \mathcal{F}(\mathbb{L}_+) \times \Theta_{I_+}} w^{(I_+, \theta)}(Z) 1_{I_+}(\ell), \quad (17)$$

$$p^{(\ell)}(x) = \frac{1}{r^{(\ell)}} \sum_{(I_+, \theta) \in \mathcal{F}(\mathbb{L}_+) \times \Theta_{I_+}} w^{(I_+, \theta)}(Z) 1_{I_+}(\ell) p^{(\theta)}(x, \ell), \quad (18)$$

where  $\Theta_{I_+}$  denotes the space of mapping  $\theta : I_+ \rightarrow \{0, 1, \dots, |Z|\}$  and,

$$w^{(I_+, \theta)}(Z) \propto w_+(I_+) [\eta_Z^{(\theta)}]^{I_+} \quad (19)$$

$$p^{(\theta)}(x, \ell | Z) = \frac{p_+(x, \ell) \psi_Z(x, \ell; \theta)}{\eta_Z^{(\theta)}(\ell)}, \quad (20)$$

$$\eta_Z^{(\theta)}(\ell) = \langle p_+(\cdot, \ell), \psi_Z(\cdot, \ell; \theta) \rangle, \quad (21)$$

$$\begin{aligned} \psi_Z(x, \ell; \theta) &= \delta_0(\theta(\ell)) q_D(x, \ell) \\ &+ (1 - \delta_0(\theta(\ell))) \frac{p_D(x, \ell) g(z_{\theta(\ell)} | x, \ell)}{\kappa(z_{\theta(\ell)})} \end{aligned} \quad (22)$$

where,  $g(z|x)$  is the single-sensor measurement likelihood,  $p_D(\cdot, \ell)$  denotes probability of detection,  $q_D(\cdot, \ell) = 1 - p_D(\cdot, \ell)$  is the probability of a missed detection, and  $\kappa(\cdot)$  intensity function of the Poisson distributed clutter process.

## 2.2. KITTI Dataset

The KITTI dataset [19] was recorded in and around Karlsruhe, Germany using a VW wagon equipped with various types of sensors including a 3D laser scanner, four video cameras and a GPS/IMU navigation system. The data was collected in both urban and suburban areas (city, rural, and highway) and information was gathered over several days. In this dataset, many of the scenes are dominated by large buildings with planar surfaces. The KITTI dataset has been widely used in mobile robotics and autonomous driving research.

## 3. Proposed Transition Model

As explained in Section 1, a transition model is required in SLAM filters to predict the states of the features at the next time step, based on the state information at the current time step. Specifically, in this study, we are interested in constructing a transition model for predicting the plane parameters at the next time step, using the current plane parameters estimated by a SLAM filter. Note that the planes involved in the transition model are expressed in the vehicle coordinate system (fixed to the vehicle), as the point cloud measurements are acquired by a laser scanner mounted on top of the experimental vehicle. Thus the plane parameters vary with time, as long as the experimental vehicle is in motion. On the other hand, if seen from the global coordinate system (fixed to the ground), the plane parameters are invariant since the planes are all static in this coordinate system.

We emphasize that the vehicle is assumed to move in a static environment. If this is not the case, we make the practical assumption that planar surfaces on other moving objects are small enough to be excluded from the set “plane observations” extracted from the 3D point cloud. That set is expected to include only relatively large planar surfaces such as walls of buildings along the road.

### 3.1. Plane Transition Model

To facilitate the design of the transition model, the two aforementioned coordinate systems are established, as shown in Figure 1. The global coordinate system  $OXYZ$  is fixed to the ground, and the vehicle coordinate system  $oxyz$  is attached to the vehicle Center of Mass and moves along with the vehicle.  $P$  is a point on a plane segmented from the 3D point cloud obtained from the on-board laser scanner.  $r_P$  and  $r$  are the position vectors of  $P$  in the global and vehicle coordinate systems, and  $r_V$  is the position vector of the vehicle centre of mass (i.e., origin  $o$ ) in the global coordinate system. The global coordinates of  $P$  are invariant since  $r_P$  is static as seen from the global coordinate system. However, the local coordinates of  $P$ , expressed in the vehicle coordinate system, vary with time due to the motion of the vehicle. As a result, the plane parameters expressed in the vehicle coordinate system

also evolve with time, and a transition model is thus needed to predict the change of plane parameters for the purpose of accurate and effective SLAM.

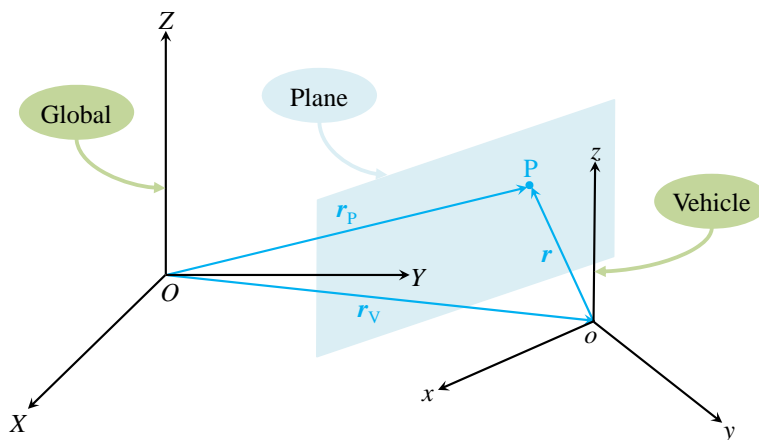


Figure 1. An example of the global and vehicle coordinate systems.

The plane equation, expressed in the vehicle coordinate system, is assumed to take the following form in this study:

$$ax + by + cz + d = 0, \tag{23}$$

where  $x, y$  and  $z$  are the coordinates of a point on the plane, and  $a, b, c$  and  $d$  are the plane parameters. Note that other forms of plane equations are also available in the literature [21], but all forms of plane equations can eventually lead to Equation (23) after simple algebraic manipulation. We may rewrite Equation (23) in the following vector form:

$$\beta^T r = 0, \tag{24}$$

where  $\beta = [a \ b \ c \ d]^T$  and  $r = [x \ y \ z \ 1]^T$ .

Assuming that the vehicle is only performing a planar motion (which is normally the case in common flat-road urban driving scenarios), the coordinates of point P in the global coordinate system  $r_P = [X \ Y \ Z \ 1]^T$ , and the coordinates of point P in the vehicle coordinate system  $r = [x \ y \ z \ 1]^T$ , are related according to the following equations describing the kinematics of the vehicle [22,23]:

$$r_P = Rr \tag{25}$$

and

$$R = \begin{bmatrix} \cos \phi & -\sin \phi & 0 & X_V \\ \sin \phi & \cos \phi & 0 & Y_V \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{26}$$

where  $\phi$  represents the heading angle of the vehicle (with respect to the  $X$  axis), and  $X_V$  and  $Y_V$  denote the coordinates of the vehicle centre of mass (origin  $o$ ) in the global coordinate system (namely  $r_V = [X_V \ Y_V \ 0 \ 1]^T$ ). The vehicle states  $\phi, X_V$  and  $Y_V$  are graphically shown in Figure 2.

Equations (25) and (26) can be rewritten as follows, for time  $k$ :

$$r_P = R_k r_k \tag{27}$$

and

$$\mathbf{R}_k = \begin{bmatrix} \cos \phi_k & -\sin \phi_k & 0 & X_{V,k} \\ \sin \phi_k & \cos \phi_k & 0 & Y_{V,k} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (28)$$

where  $\mathbf{r}_k = [x_k \ y_k \ z_k \ 1]^T$ . Similarly, we have the following equations for time  $k + 1$ :

$$\mathbf{r}_P = \mathbf{R}_{k+1} \mathbf{r}_{k+1} \quad (29)$$

and

$$\mathbf{R}_{k+1} = \begin{bmatrix} \cos \phi_{k+1} & -\sin \phi_{k+1} & 0 & X_{V,k+1} \\ \sin \phi_{k+1} & \cos \phi_{k+1} & 0 & Y_{V,k+1} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (30)$$

where  $\mathbf{r}_{k+1} = [x_{k+1} \ y_{k+1} \ z_{k+1} \ 1]^T$ .

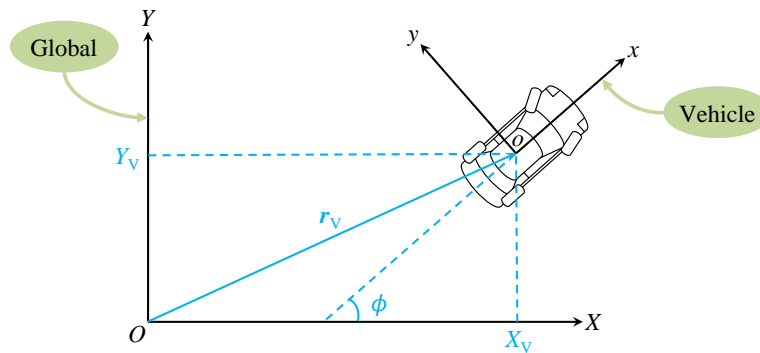


Figure 2. Top view of the global and vehicle coordinate systems.

The vector  $\mathbf{r}_P$  is invariant with time, since it is the position vector of P in the global coordinate system. As a result, one can achieve the following equation by combining Equations (27) and (29):

$$\mathbf{r}_{k+1} = \mathbf{R}_{k+1}^{-1} \mathbf{R}_k \mathbf{r}_k. \quad (31)$$

Note that Equation (24) can be rewritten as follows, for time  $k$ :

$$\boldsymbol{\beta}_k^T \mathbf{M}_k^{-1} \mathbf{M}_k \mathbf{r}_k = 0, \quad (32)$$

where  $\mathbf{M}_k$  represents a 4-by-4 invertible matrix. We might as well let  $\mathbf{M}_k = \mathbf{R}_{k+1}^{-1} \mathbf{R}_k$  and combine Equations (31) and (32), then we arrive at:

$$\boldsymbol{\beta}_k^T \mathbf{M}_k^{-1} \mathbf{r}_{k+1} = 0. \quad (33)$$

The plane Equation (24) for time  $k + 1$  can be expressed as follows:

$$\boldsymbol{\beta}_{k+1}^T \mathbf{r}_{k+1} = 0. \quad (34)$$

Combining Equations (33) and (34) leads to the following transition model for plane parameters:

$$\boldsymbol{\beta}_{k+1}^T = \boldsymbol{\beta}_k^T \mathbf{M}_k^{-1}, \quad (35)$$

where  $\mathbf{M}_k^{-1} = (\mathbf{R}_{k+1}^{-1} \mathbf{R}_k)^{-1} = \mathbf{R}_k^{-1} \mathbf{R}_{k+1}$ .

This plane transition model indicates that the plane parameters at time  $k + 1$  can be calculated based on the plane parameter information at time  $k$  and a vehicle-motion-dependent 4-by-4 matrix  $M_k^{-1}$ . Note that the computation of matrix  $M_k^{-1} = R_k^{-1}R_{k+1}$  requires the inverse of  $R_k$ . It can be easily proven that the matrix  $R_k$ , expressed by Equation (28), has full rank and its inverse  $R_k^{-1}$  exists at all times. Thus, the matrix  $M_k^{-1}$  can be computed as long as  $R_{k+1}$  is available. However, this matrix  $R_{k+1}$  (as indicated by Equation (30)) requires the vehicle state information at the future time  $k + 1$ , which is not available in real-time SLAM applications. To tackle this issue, in the following section we will introduce a new vehicle transition model and demonstrate how the vehicle states at time  $k + 1$  can be predicted.

### 3.2. Vehicle Transition Model

As mentioned above, the transformation matrix  $R_{k+1}$  requires the unknown information from the future time  $k + 1$ . In order to obtain matrix  $R_{k+1}$ , in this section a vehicle transition model is introduced to predict the future vehicle states at time  $k + 1$ , using the available information at time  $k$ .

In Bayesian filtering, the Constant Turn (CT) model [24–26] which describes the maneuver of the vehicle motion is commonly used. This model is formulated based on the assumption that the vehicle maneuvers with a (nearly) constant speed and a (nearly) constant angular rate [24]. Based on the CT model, an Extended Constant Turn (ECT) model is proposed in this work, which can be mathematically expressed as follows:

$$\mathbf{x}_{k+1} = \mathbf{F}_k \mathbf{x}_k + \mathbf{G} \mathbf{u}, \quad (36)$$

with

$$\begin{aligned} \mathbf{x}_{k+1} &= [X_{V,k+1} \dot{X}_{V,k+1} Y_{V,k+1} \dot{Y}_{V,k+1} \phi_{k+1} \dot{\phi}_{k+1}]^T, \\ \mathbf{x}_k &= [X_{V,k} \dot{X}_{V,k} Y_{V,k} \dot{Y}_{V,k} \phi_k \dot{\phi}_k]^T, \\ \mathbf{u} &= [u_X u_Y u_\phi]^T, \\ \mathbf{F}_k &= \begin{bmatrix} 1 & \frac{\sin(\dot{\phi}_k T)}{\dot{\phi}_k} & 0 & \frac{\cos(\dot{\phi}_k T) - 1}{\dot{\phi}_k} & 0 & 0 \\ 0 & \cos(\dot{\phi}_k T) & 0 & -\sin(\dot{\phi}_k T) & 0 & 0 \\ 0 & \frac{1 - \cos(\dot{\phi}_k T)}{\dot{\phi}_k} & 1 & \frac{\sin(\dot{\phi}_k T)}{\dot{\phi}_k} & 0 & 0 \\ 0 & \sin(\dot{\phi}_k T) & 0 & \cos(\dot{\phi}_k T) & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \text{ and } \mathbf{G} = \begin{bmatrix} \frac{T^2}{2} & 0 & 0 \\ T & 0 & 0 \\ 0 & \frac{T^2}{2} & 0 \\ 0 & T & 0 \\ 0 & 0 & \frac{T^2}{2} \\ 0 & 0 & T \end{bmatrix}, \end{aligned}$$

where  $X_{V,k}$ ,  $Y_{V,k}$ ,  $X_{V,k+1}$  and  $Y_{V,k+1}$  denote the vehicle longitudinal and lateral displacements (i.e., the coordinates of the vehicle centre of mass as introduced in Section 3.1) in the global coordinate system at time  $k$  and  $k + 1$ ,  $\phi_k$  and  $\phi_{k+1}$  represent the vehicle heading angles at time  $k$  and  $k + 1$ ,  $u_X$ ,  $u_Y$  and  $u_\omega$  are the vehicle velocity and yaw rate noise components, and  $T$  stands for the sampling period. By means of this vehicle transition model (36), the required vehicle state information at time  $k + 1$  can be predicted and in turn the matrices  $R_{k+1}$  and  $M_k^{-1}$  can be calculated.

Note that the plane transition model proposed in Section 3.1 (i.e., Equation (35)), along with the above ECT vehicle transition model, constitutes a complete state transition model that propagates the state densities of the planar features to the next time. This state transition model plays a key role in the prediction step for prospective planar-feature-based SLAM filters.

## 4. Implementation

In this section, we introduce the Sequential Monte Carlo (SMC) implementation of the proposed transition model, for the recently developed LMB filter. The LMB filter has been used in many applications such as target tracking [27,28], SLAM [29], visual tracking [30], and sensor control [31,32]. In the SMC implementation, the LMB posterior for the map features (planes) at each time step are



represented by a set of particles. These particles are propagated using the proposed transition model to obtain the LMB posterior at the next time step.

Assuming the following LMB posterior  $\pi_k$  at time  $k$ :

$$\pi_k = \left\{ (r_k^{(\ell)}, p_k^{(\ell)}) \right\}_{\ell \in \mathbb{L}}, \quad (37)$$

where  $\ell$  represents the label of a plane,  $r_k^{(\ell)}$  denotes the existence probability of the plane  $\ell$ ,  $p_k^{(\ell)}$  is the spatial distribution of its parameters, and  $\mathbb{L}$  stands for the label space. In an SMC implementation, the spatial distribution is represented by a set of weighted samples [28], namely:

$$p_k^{(\ell)} = \sum_{i=1}^{N^{(\ell)}} \omega_i^{(\ell)} \delta(a - a_{i,k}^{(\ell)}) \delta(b - b_{i,k}^{(\ell)}) \delta(c - c_{i,k}^{(\ell)}) \delta(d - d_{i,k}^{(\ell)}), \quad (38)$$

where  $N^{(\ell)}$  represents the number of particles,  $\omega_i^{(\ell)}$  denotes the weight of the  $i$ -th particle, and  $\delta$  is the Dirac delta function.

For each particle  $\beta_{i,k}^{(\ell)} = [a_{i,k}^{(\ell)}, b_{i,k}^{(\ell)}, c_{i,k}^{(\ell)}, d_{i,k}^{(\ell)}]^T$  at time  $k$ , we are able to calculate the corresponding predicted particle  $\beta_{i,k+1}^{(\ell)} = [a_{i,k+1}^{(\ell)}, b_{i,k+1}^{(\ell)}, c_{i,k+1}^{(\ell)}, d_{i,k+1}^{(\ell)}]^T$  at time  $k + 1$  based on the transition model proposed in Section 3. As a result, after the prediction stage, we will end up with a new set of particles with the same weights, namely:

$$\left\{ \omega_{i,k+1}^{(\ell)}, \beta_{i,k+1}^{(\ell)} \right\}_{i=1}^{N^{(\ell)}},$$

where  $\omega_{i,k+1}^{(\ell)} = \omega_{i,k}^{(\ell)}$ . Then, based on the new predicted particles at time  $k + 1$ , the plane parameter estimate  $\hat{\beta}_{k+1}^{(\ell)}$  can be achieved from the distribution particles as the Expected A Posteriori (EAP) estimate (i.e., weighted average), or the Maximum A Posteriori (MAP) estimate (i.e., considering the particle with the largest weight as the estimate).

The above implementation procedure is illustrated step by step in Algorithm 1, in the form of a pseudo-code.

---

#### Algorithm 1 Step-by-Step Implementation of the Proposed Transition Model

---

**Require:** vehicle states  $X_{V,k}$ ,  $Y_{V,k}$ ,  $\phi_k$ , and particles  $\left\{ \omega_{i,k}^{(\ell)}, \beta_{i,k}^{(\ell)} \right\}_{i=1}^{N^{(\ell)}}$  representing the distribution of each plane with label  $\ell$ , for all labels  $\ell \in \mathbb{L}$

- 1: compute matrix  $R_k$  ▷ use Equation (28)
- 2: generate a noise sample  $\mathbf{u} \sim \mathcal{N}(0, \Sigma)$  ▷  $\Sigma \sim \text{diag}(\text{noise variances})$
- 3: generate  $X_{V,k+1}$ ,  $Y_{V,k+1}$  and  $\phi_{k+1}$  ▷ use Equation (36)
- 4: compute matrix  $R_{k+1}$  ▷ use Equation (30)
- 5:  $M_k^{-1} \leftarrow R_k^{-1} R_{k+1}$
- 6: **for**  $\ell \in \mathbb{L}$  **do**
- 7:   **for**  $i \leq N^{(\ell)}$  **do**
- 8:      $\beta_{i,k+1}^{(\ell)T} \leftarrow \beta_{i,k}^{(\ell)T} \times M_k^{-1}$
- 9:      $\omega_{i,k+1}^{(\ell)} \leftarrow \omega_{i,k}^{(\ell)}$
- 10:   **end for**
- 11:    $\hat{\beta}_{k+1}^{(\ell)} \leftarrow \sum_{i=1}^{N^{(\ell)}} \omega_{i,k+1}^{(\ell)} \beta_{i,k+1}^{(\ell)}$  ▷ compute the EAP estimate
- 12: **end for**

---

## 5. Simulation Results

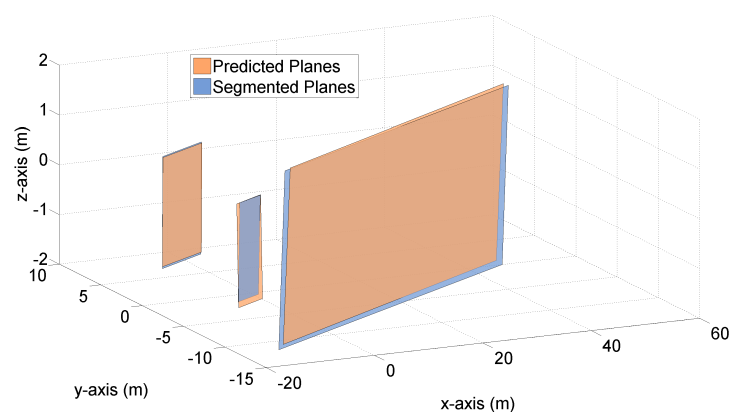
In this section, we demonstrate how well the proposed transition model performs, compared with the actually measured results from the KITTI dataset. Specifically, we generate the predicted planes at time  $k + 1$  using the plane information at time  $k$ , by means of the proposed transition model, and we segment planes from the point cloud data in the KITTI dataset at time  $k + 1$ , by means of the Modified Selective Statistical Estimator (MSSE) [33]. Then, the predicted planes at time  $k + 1$  are compared with the segmented planes at time  $k + 1$ , and both graphical and numerical results are demonstrated to show the closeness of the predicted planes and the segmented planes.

The point cloud data from the folder “2011\_09\_26\_drive\_0005\_sync” in the KITTI dataset are employed in the simulation studies for plane segmentation. The details for accessing and using the KITTI dataset are available in [19]. The folder “2011\_09\_26\_drive\_0005\_sync” includes 154 time steps (i.e.,  $k \in [0, 153]$ ), and the points obtained at  $k = 11$ ,  $k = 12$ ,  $k = 124$  and  $k = 125$  are used in the simulation. The number of points at  $k = 11$ ,  $k = 12$ ,  $k = 124$  and  $k = 125$  are 123,334, 123,316, 118,950 and 118,572 respectively.

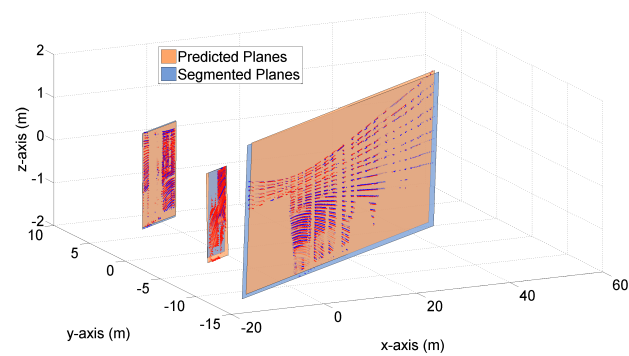
### 5.1. Graphical Results

Figures 3 and 4 demonstrate both the predicted planes and the segmented planes at time  $k = 125$  for one Monte Carlo (MC) run. The predicted planes are in orange color, and are generated by the proposed transition model using the LIDAR measurements and vehicle states at time  $k = 124$ . The segmented planes are in blue color, and are plotted based on the point cloud measurements at time  $k = 125$ . Apart from the planes themselves, the predicted and measured point clouds at time  $k = 125$  are also plotted in Figure 4, on top of the corresponding planes.

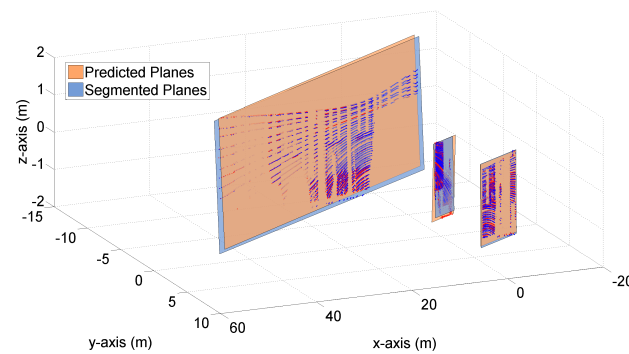
Figure 3 shows that by means of the transition model and the available information at time  $k = 124$ , three planes are predicted to exist at time  $k = 125$ . Also, three segmented planes are present in this figure based on the LIDAR measurements at time  $k = 125$ . It is clearly observed in Figures 3 that each predicted plane is associated with one segmented plane. Namely, these six planes constitute three pairs of planes, with each pair composed of one predicted plane and one segmented plane. Besides, we see that for each pair of orange and blue planes, the areas of planes are fairly close and the distances between each pair of plane vertexes are rather small (‘Each pair of plane vertexes’ is referred to one vertex on the predicted plane and its closest counterpart on the segmented plane).



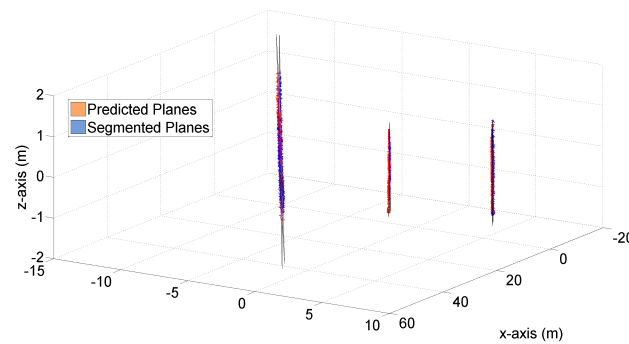
**Figure 3.** Three pairs of predicted and segmented planes at time  $k = 125$ .



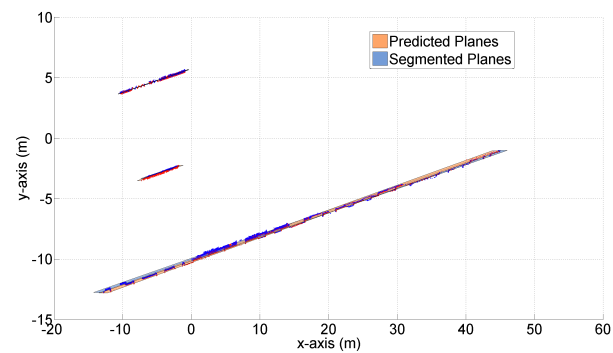
(a) Angle 1



(b) Angle 2



(c) Angle 3



(d) Angle 4

**Figure 4.** Multi-angle comparison of the predicted and segmented planes at time  $k = 125$ .

Figure 4 demonstrates the three pairs of planes from different angles. Again, we observe the closeness of the predicted and segmented planes in terms of plane areas and vertex distances, when

viewed from different angles. Figure 4c shows a ‘side view’ of the three pairs of planes, from which we clearly see that the predicted and segmented planes (and their normal vectors) almost coincide. Similar results are also observed in Figure 4d where a ‘top view’ of the planes is shown.

In addition to  $k = 125$ , our simulation studies have included a large range of other time steps. Figures 5 and 6 provide the results of another time step  $k = 12$ . The rest of time steps present similar results and are not shown in this paper for the purpose of brevity. In the following section, the above graphical results will be further supplemented and clarified, by means of detailed numerical results.

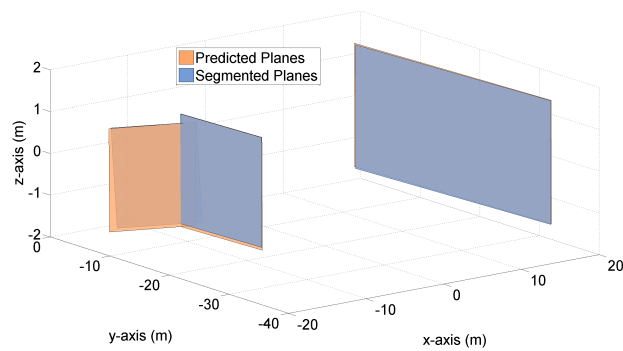
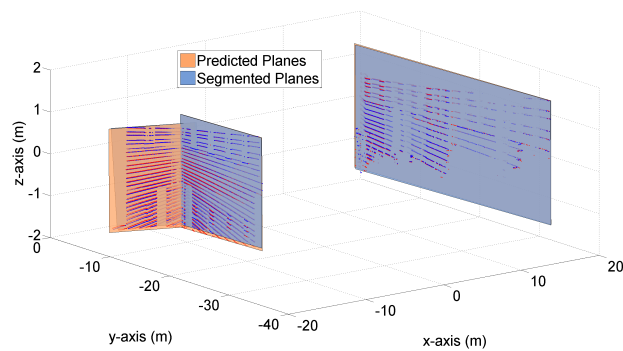
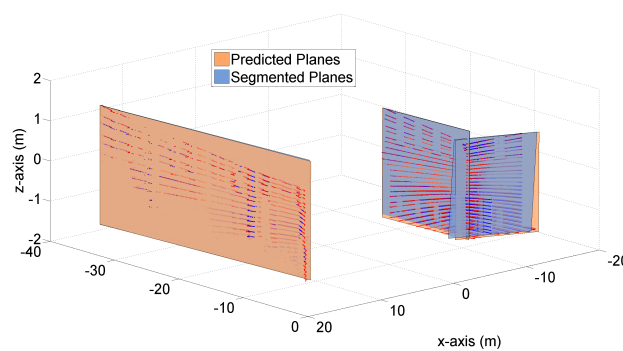


Figure 5. Three pairs of predicted and segmented planes at time  $k = 12$ .

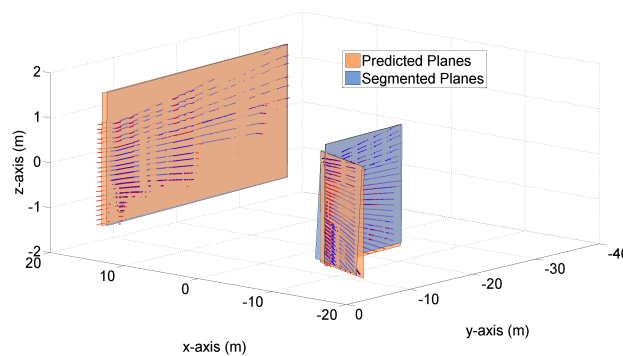


(a) Angle 1

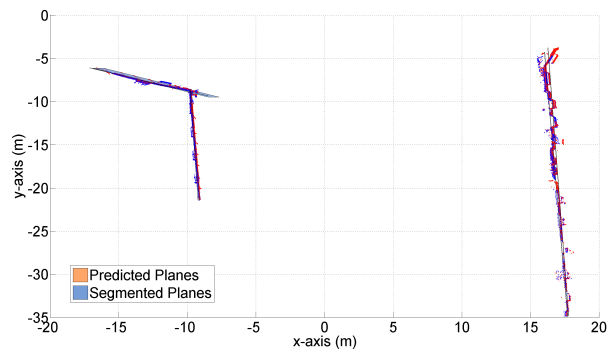


(b) Angle 2

Figure 6. Cont.



(c) Angle 3



(d) Angle 4

**Figure 6.** Multi-angle comparison of the predicted and segmented planes at time  $k = 12$ .

## 5.2. Numerical Results

In Figures 3–6, we have graphically demonstrated the comparison results between the predicted and segmented planes, in terms of three important indicators – plane area, vertex distance and normal vector. In this section, we provide the results of our numerical simulation for 100 MC runs, in order to demonstrate ‘how close’ exactly these planes are.

The first two indicators, plane area and vertex distance, are mainly used to quantify the closeness in terms of plane dimensions (sizes). Table 1 shows the areas of the predicted and segmented planes for time step  $k = 125$  (averaged from 100 MC runs), and Table 2 gives the distances between each pair of plane vertexes for time step  $k = 125$  (averaged from 100 MC runs). We see in these tables that the areas of the predicted and segmented planes are close, and the distances between each pair of vertexes are small. These numerical results are consistent with the graphical results shown in Section 5.1.

**Table 1.** Areas of the predicted and segmented planes for time step  $k = 125$ .

	Plane Pair 1	Plane Pair 2	Plane Pair 3
Predicted Plane	13.7147 m <sup>2</sup>	219.9937 m <sup>2</sup>	22.9218 m <sup>2</sup>
Segmented Plane	11.0470 m <sup>2</sup>	226.3712 m <sup>2</sup>	23.2926 m <sup>2</sup>

**Table 2.** Distances between each pair of plane vertexes for time step  $k = 125$ .

	Plane Pair 1	Plane Pair 2	Plane Pair 3
Vertex Pair 1	0.2503 m	0.7089 m	0.0603 m
Vertex Pair 2	0.8519 m	0.8667 m	0.0294 m
Vertex Pair 3	0.1293 m	0.9060 m	0.0837 m
Vertex Pair 4	0.8107 m	1.0730 m	0.0677 m

The third indicator, normal vector, is employed to evaluate the closeness in terms of plane orientation. Table 3 shows the angles between each pair of normal vectors for time step  $k = 125$  (averaged from 100 MC runs) ('Each pair of normal vectors' is referred to the normal vector on the predicted plane and its counterpart on the segmented plane). The small magnitudes of these angles imply that the predicted and segmented planes are very close in terms of orientation. This explains the coincidence of the predicted and segmented planes seen in Figure 4c,d.

**Table 3.** Angles between each pair of normal vectors for time step  $k = 125$ .

	Plane Pair 1	Plane Pair 2	Plane Pair 3
Angle Between Normal Vectors	0.0654 rad/3.7458°	0.0129 rad/0.7382°	0.0037 rad/0.2115°

Besides the above three indicators for closeness evaluation, the execution time of the program is also recorded. The proposed transition model is implemented based on Algorithm 1, and the program is executed on a laptop equipped with an Intel i7-5600U CPU and an 8G RAM. For time step  $k = 125$ , the execution time is 0.0375 s (averaged from 100 MC runs).

Apart from  $k = 125$ , the numerical results for another time step  $k = 12$  are also presented in Tables 4–6. Similarly, these results show that the predicted planes (obtained using the information at  $k = 11$ ) are fairly close to the segmented planes at  $k = 12$ , in terms of plane areas, vertex distances and normal vectors. Besides, the execution time of the program for  $k = 12$  is 0.0369 s (averaged from 100 MC runs). For the purpose of brevity, the numerical results for other time steps are omitted in this paper.

**Table 4.** Areas of the predicted and segmented planes for time step  $k = 12$ .

	Plane Pair 1	Plane Pair 2	Plane Pair 3
Predicted Plane	34.5121 m <sup>2</sup>	22.7771 m <sup>2</sup>	91.5893 m <sup>2</sup>
Segmented Plane	33.7797 m <sup>2</sup>	23.0555 m <sup>2</sup>	90.5685 m <sup>2</sup>

**Table 5.** Distances between each pair of plane vertexes for time step  $k = 12$ .

	Plane Pair 1	Plane Pair 2	Plane Pair 3
Vertex Pair 1	0.1311 m	0.8031 m	0.1053 m
Vertex Pair 2	0.1117 m	0.1600 m	0.1119 m
Vertex Pair 3	0.0602 m	0.2778 m	0.5599 m
Vertex Pair 4	0.0924 m	1.0056 m	0.5558 m

**Table 6.** Angles between each pair of normal vectors for time step  $k = 12$ .

	Plane Pair 1	Plane Pair 2	Plane Pair 3
Angle Between Normal Vectors	0.0034 rad/0.1936°	0.1133 rad/6.4912°	0.0070 rad/0.3987°

## 6. Conclusions

The majority of the current statistical SLAM solutions are based on using point features such as the representation of landmarks. The fast advancement of sensory technology makes it possible to utilize more sophisticated features in SLAM applications to achieve superior results while lowering computational cost. This paper presented the idea of using planar features for statistical SLAM, and proposed a stochastic transition model to propagate the plane parameters to the next time. A large range of simulation studies using real-world measurements have been conducted to evaluate the proposed transition model. Both graphical and numerical results show that the predicted planes

generated by the proposed transition model closely match the segmented planes resulting from real-world point cloud measurements, in terms of three important indicators; plane area, vertex distance and normal vector. In the next step, we will look into other planar-feature-based stochastic transition models, and investigate the application of such transition models in statistical SLAM tasks.

**Author Contributions:** Conceptualization, R.H. and A.K.G.; Methodology, C.F., M.I.H. and R.T.; Software, C.F. and W.C.; Validation, C.F. and W.C.; Formal Analysis, C.F. and R.H.; Resources, A.K.G. and C.F.; Visualization, C.F. and R.T.; Supervision, R.H. and A.B.H.; Project Administration, C.F. and R.H.; Funding Acquisition, C.F. and R.H.

**Funding:** This research was funded by the National Natural Science Foundation of China (Grant No. 51805055), the National Key Research and Development Project (Grant No. 2018YFB0106100), and the Australian Research Council (the ARC Linkage Project grant LP160101081).

**Conflicts of Interest:** The authors declare no conflict of interest. The funding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

## Abbreviations

The following abbreviations are used in this manuscript:

SLAM	Simultaneous Localization and Mapping
EKF	Extended Kalman Filter
RFS	Random Finite Set
LMB	Labeled Multi-Bernoulli
CT	Constant Turn
ECT	Extended Constant Turn
SMC	Sequential Monte Carlo
EAP	Expected A Posteriori
MAP	Maximum A Posteriori
MSSE	Modified Selective Statistical Estimator
MC	Monte Carlo

## References

1. Cadena, C.; Carlone, L.; Carrillo, H.; Latif, Y.; Scaramuzza, D.; Neira, J.; Reid, I.D.; Leonard, J.J. Simultaneous Localization and Mapping: Present, Future, and the Robust-Perception Age. *IEEE Trans. Robot.* **2016**, *32*, 1309–1332. [[CrossRef](#)]
2. Mullane, J.; Vo, B.; Adams, M.D.; Vo, B. A Random-Finite-Set Approach to Bayesian SLAM. *IEEE Trans. Robot.* **2011**, *27*, 268–282. [[CrossRef](#)]
3. Davison, A.J. Real-Time Simultaneous Localisation and Mapping with a Single Camera. In Proceedings of the Ninth IEEE International Conference on Computer Vision, Nice, France, 13–16 October 2003; IEEE Computer Society: Washington, DC, USA, 2003; pp. 1403–1410.
4. Thrun, S.; Koller, D.; Ghahramani, Z.; Durrant-Whyte, H.; Ng, A.Y. Simultaneous Mapping and Localization with Sparse Extended Information Filters: Theory and Initial Results. In *Algorithmic Foundations of Robotics V*; Springer Tracts in Advanced Robotics; Springer: Berlin, Germany, 2002; Volume 7, pp. 363–380.
5. Montemerlo, M.; Thrun, S.; Koller, D.; Wegbreit, B. *FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem*; AAAI Press/The MIT Press: Cambridge, MA, USA, 2002; pp. 593–598.
6. Rozsa, Z.; Sziranyi, T. Obstacle prediction for automated guided vehicles based on point clouds measured by a tilted LIDAR sensor. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 2708–2720. [[CrossRef](#)]
7. Fossel, J.; Hennes, D.; Claes, D.; Alers, S.; Tuyls, K. OctoSLAM: A 3D mapping approach to situational awareness of unmanned aerial vehicles. In Proceedings of the 2013 International Conference on Unmanned Aircraft Systems (ICUAS), Atlanta, GA, USA, 28–31 May 2013; pp. 179–188.
8. Im, J.; Im, S.; Jee, G. Extended line map-based precise vehicle localization using 3D LIDAR. *Sensors* **2018**, *18*, 3179. [[CrossRef](#)] [[PubMed](#)]
9. Ahn, S.; Chung, W.K. Efficient SLAM algorithm with hybrid visual map in an indoor environment. In Proceedings of the 2007 International Conference on Control, Automation and Systems, Seoul, Korea, 17–20 October 2007; pp. 663–667.



10. Chen, Z.; Samarabandu, J.; Rodrigo, R. Recent advances in simultaneous localization and map-building using computer vision. *Adv. Robot.* **2007**, *21*, 233–265. [[CrossRef](#)]
11. Tomono, M. Robust 3D SLAM with a stereo camera based on an edge-point ICP algorithm. In Proceedings of the 2009 IEEE International Conference on Robotics and Automation, Kobe, Japan, 12–17 May 2009; pp. 4306–4311.
12. Sun, F.; Zhou, Y.; Li, C.; Huang, Y. Research on active SLAM with fusion of monocular vision and laser range data. In Proceedings of the 2010 8th World Congress on Intelligent Control and Automation, Jinan, China, 7–9 July 2010; pp. 6550–6554.
13. Henry, P.; Krainin, M.; Herbst, E.; Ren, X.; Fox, D. RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments. *Int. J. Robot. Res.* **2012**, *31*, 647–663. [[CrossRef](#)]
14. Endres, F.; Hess, J.; Engelhard, N.; Sturm, J.; Cremers, D.; Burgard, W. An evaluation of the RGB-D SLAM system. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation, Saint Paul, MN, USA, 14–18 May 2012; pp. 1691–1696.
15. Fazli, S.; Kleeman, L. Simultaneous landmark classification, localization and map building for an advanced sonar ring. *Robotica* **2006**, *25*, 283–296. [[CrossRef](#)]
16. Kim, P.; Coltin, B.; Kim, H.J. Linear RGB-D SLAM for Planar Environments. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Heidelberg, Germany, 2018; Volume 11208 LNCS, pp. 350–366.
17. Grant, W.S.; Voorhies, R.C.; Itti, L. Efficient Velodyne SLAM with point and plane features. In *Autonomous Robots*; Springer: New York, NY, USA, 2018.
18. Pietzsch, T. Planar features for visual SLAM. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Heidelberg, Germany, 2008; Volume 5243 LNAI, pp. 119–126.
19. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The KITTI dataset. *Int. J. Robot. Res.* **2013**, *32*, 1231–1237. [[CrossRef](#)]
20. Reuter, S.; Vo, B.T.; Vo, B.N.; Dietmayer, K. The labeled multi-Bernoulli filter. *IEEE Trans. Signal Process.* **2014**, *62*, 3246–3260.
21. Hoseinnezhad, R.; Bab-Hadiashar, A. An M-estimator for high breakdown robust estimation in computer vision. *Comput. Vis. Image Underst.* **2011**, *115*, 1145–1156. [[CrossRef](#)]
22. Jazar, R.N. *Vehicle Dynamics: Theory and Application*; Springer: Berlin, Germany, 2014; Chapter 5, pp. 233–312.
23. Jazar, R.N. *Advanced Dynamics: Rigid Body, Multibody, and Aerospace Applications*; Wiley: Hoboken, NJ, USA, 2011; Chapter 4, pp. 357–415.
24. Li, X.R.; Jilkov, V.P. Survey of maneuvering target tracking. Part I. Dynamic models. *IEEE Trans. Aerosp. Electron. Syst.* **2003**, *39*, 1333–1364.
25. Gostar, A.K.; Hoseinnezhad, R.; Bab-Hadiashar, A.; Liu, W. Sensor-Management for Multitarget Filters via Minimization of Posterior Dispersion. *IEEE Trans. Aerosp. Electron. Syst.* **2017**, *53*, 2877–2884. [[CrossRef](#)]
26. Gostar, A.K.; Hoseinnezhad, R.; Bab-Hadiashar, A. Sensor control for multi-object tracking using labeled multi-Bernoulli filter. In Proceedings of the 17th International Conference on Information Fusion (FUSION), Salamanca, Spain, 7–10 July 2014; pp. 1–8.
27. Li, S.; Yi, W.; Hoseinnezhad, R.; Wang, B.; Kong, L. Multiobject Tracking for Generic Observation Model Using Labeled Random Finite Sets. *IEEE Trans. Signal Process.* **2018**, *66*, 368–383. [[CrossRef](#)]
28. Reuter, S. Multi-Object Tracking Using Random Finite Sets. Ph.D. Thesis, Ulm University, Ulm, Germany, 2014.
29. Deusch, H.; Reuter, S.; Dietmayer, K. The Labeled Multi-Bernoulli SLAM Filter. *IEEE Signal Process. Lett.* **2015**, *22*, 1561–1565. [[CrossRef](#)]
30. Hossain, M.I.; Gostar, A.K.; Bab-Hadiashar, A.; Hoseinnezhad, R. Visual Mitosis Detection and Cell Tracking Using Labeled Multi-Bernoulli Filter. In Proceedings of the 21th International Conference on Information Fusion (FUSION), Cambridge, UK, 10–13 July 2018; pp. 2468–2472.
31. Wang, X.; Hoseinnezhad, R.; Gostar, A.K.; Rathnayake, T.; Xu, B.; Bab-Hadiashar, A. Multi-sensor control for multi-object Bayes filters. *Signal Process.* **2018**, *142*, 260–270. [[CrossRef](#)]
32. Gostar, A.K.; Hoseinnezhad, R.; Rathnayake, T.; Wang, X.; Bab-Hadiashar, A. Constrained Sensor Control for Labeled Multi-Bernoulli Filter Using Cauchy-Schwarz Divergence. *IEEE Signal Process. Lett.* **2017**, *24*, 1313–1317. [[CrossRef](#)]



33. Bab-Hadiashar, A.; Suter, D. Robust segmentation of visual data using ranked unbiased scale estimate. *Robotica* **1999**, *17*, 649–660. [[CrossRef](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).