sensors

MDPI

Article

Rapid Motion Segmentation of LiDAR Point Cloud Based on a Combination of Probabilistic and Evidential Approaches for Intelligent Vehicles

Kichun Jo [1,*], Sumyeong Lee [2], Chansoo Kim [2] and Myoungho Sunwoo [2]

[1] Department of Smart Vehicle Engineering, Konkuk University, Seoul 05029, Korea
[2] Department of Automotive Engineering, Hanyang University, Seoul 04763, Korea; smlee950728@gmail.com (S.L.); chansoo7857@gmail.com (C.K.); msunwoo728@gmail.com (M.S.)
* Correspondence: kichun@konkuk.ac.kr; Tel.: +82-2-2049-6265

Received: 23 August 2019; Accepted: 18 September 2019; Published: 23 September 2019

Abstract: Point clouds from light detecting and ranging (LiDAR) sensors represent increasingly important information for environmental object detection and classification of automated and intelligent vehicles. Objects in the driving environment can be classified as either *dynamic* or *static* depending on their movement characteristics. A LiDAR point cloud is also segmented into *dynamic* and *static* points based on the motion properties of the measured objects. The segmented motion information of a point cloud can be useful for various functions in automated and intelligent vehicles. This paper presents a fast motion segmentation algorithm that segments a LiDAR point cloud into *dynamic* and *static* points in real-time. The segmentation algorithm classifies the motion of the latest point cloud based on the LiDAR's laser beam characteristics and the geometrical relationship between consecutive LiDAR point clouds. To accurately and reliably estimate the motion state of each LiDAR point considering the measurement uncertainty, both probability theory and evidence theory are employed in the segmentation algorithm. The probabilistic and evidential algorithm segments the point cloud into three classes: *dynamic*, *static*, and *unknown*. Points are placed in the *unknown* class when LiDAR point cloud is not sufficient for motion segmentation. The point motion segmentation algorithm was evaluated quantitatively and qualitatively through experimental comparisons with previous motion segmentation methods.

Keywords: LiDAR; laser beam model; point motion classification; Dempster-Sharfer theory; intelligent vehicle

1. Introduction

LiDAR systems are rapidly becoming an integral part of automated and intelligent vehicles for environmental awareness. The price of LiDAR sensors is reducing, and automakers are increasingly installing LiDAR in production vehicles for advanced intelligent functions [1,2]. LiDAR measures the distance and direction of the surrounding environment by emitting laser pulses in certain directions and measuring the time-of-flight (ToF) of each laser pulse reflected by the environment. The directions and distances can be converted to a digital 3D representation called a point cloud to express the spatial information of the surrounding environment. Because LiDAR uses light waves, the measured point cloud can represent spatial information very accurately. In addition, LiDAR point clouds can be fused with data from other sensors, such as radars, and cameras, to gain more meaningful information about the vehicle's environment.

All objects in the driving environment are classified as dynamic or static according to the moving conditions. Therefore, the LiDAR point cloud from detected objects can also be classified as dynamic or static point-wise depending on the motion state of the object. Such point-wise classification of point cloud states can be used for safety and convenience functions in automated and intelligent vehicles. For instance, points classified as static are measured from the surfaces of static objects, such as curbs, poles, buildings, and parked vehicles. Such a static point cloud can be applied to various automated and intelligent driving functions, such as mapping, localization, and collision avoidance systems [3–5]. Points classified as dynamic are detected from objects that have speeds above a certain level, such as nearby moving vehicles, motorcycles, and pedestrians. These points can be used for object tracking or motion prediction, which are necessary functions for automated and intelligent vehicles for tasks such as autonomous emergency braking (AEB), lane keeping, traffic jam assistance, and adaptive cruise control (ACC) systems.

As shown in the previous examples, the information of the point cloud state is classified according to the motion is useful for automated and intelligent vehicles. This paper proposes an algorithm to rapidly segment the motion states of a point cloud detected by LiDAR in real-time. The overall process of the proposed algorithm is shown in Figure 1. The rapid motion segmentation algorithm has inputs of LiDAR's 3D point cloud and the 3D pose (position and direction) of the LiDAR sensor. The sensor pose can be estimated from an inertial measurement unit (IMU) or the vehicle's on-board motion sensors (such as wheel speed sensors or steering angle sensor). Then, point motion segmentation is performed by applying the laser beam characteristics to the pose correlation between consecutive LiDAR point clouds. A combination of probability theory and evidence theory is applied to accurately and reliably update the motion state of points. The algorithm performs point-wise segmentation of the point cloud into three states: *dynamic*, *static*, and *unknown*. *dynamic* information is detected from an object moving above a certain speed, and *static* information is detected from a stationary object. If there are insufficient consecutive LiDAR point clouds for motion classification, some points are classified as *unknown*. The performance of the proposed algorithm was evaluated quantitatively and qualitatively through comparison with existing methods.

This research has three main contributions: (1) reflecting the laser characteristics of LiDAR, (2) applying a combination of probabilistic and evidential approaches to update the motion state of points, and (3) online motion updated for real-time applications. This paper focuses on the characteristics of lasers, such as multi-echo, beam divergence, and horizontal and vertical resolution, so that it can segment the motion of points more accurately than existing algorithms, such as occupancy grid mapping. In addition, when updating the state information, a combination of probabilistic and evidential modeling is applied to more accurately reflect the actual LiDAR characteristics to update motion in a point-wise manner. Because all the proposed updating processes are real-time, they are suitable for real-time application in automated and intelligent vehicle systems.

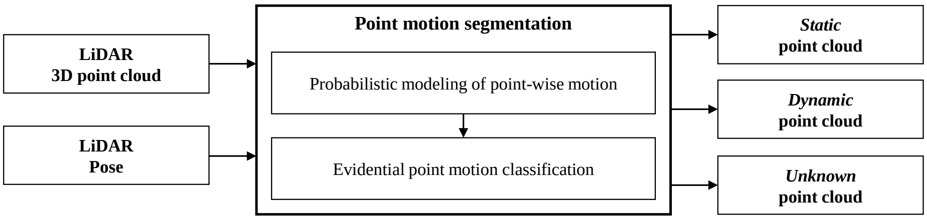

LiDAR
3D point cloud
LiDAR
Pose
Point motion segmentation
Probabilistic modeling of point-wise motion
Evidential point motion classification
Static
point cloud
Dynamic
point cloud
Unknown
point cloud

Figure 1. System architecture of the rapid point motion segmentation algorithm.

2. Previous Studies

With the development of autonomous vehicles, LiDAR is being more widely used, and many studies of LiDAR are accordingly being conducted. However, there have not been many studies that classify the point-wise motion of LiDAR measurements themselves in real-time. In the field of autonomous and intelligent vehicle systems, studies related to the proposed algorithm aim to generate static environment maps or remove nonstatic points based on tracking results.

An occupancy grid map is a typical method for generating a static environment map using LiDAR measurements. The occupancy grid map divides the environment around the vehicle into a 2D grid or 3D voxel cells with uniform size. The occupancy level of each cell can be updated through the laser′s ray tracing. The occupancy level of a grid (or voxel) cell passing by the laser becomes lower because the physical space of the cell is likely to be free. Conversely, the levels of cells located on the reflecting surface become higher. Based on these principles, static objects are classified as occupied when the occupancy level exceeds a certain threshold. Contrarily, for a moving object, the occupancy level of the cell is not constantly accumulated, so it is not classified as occupied. The numerical value of each cell's occupancy level is updated based on probability theory [6] or belief theory [7]. Static maps in large-scale traffic environment are constructed using LiDAR sensors with probabilistic and belief approaches [8–10]. Moras et al. presented an occupancy grid framework that generates a global static map and classifies local moving objects simultaneously [11–13]. Classification of traffic objects (such as vehicles, pedestrians, road curbs, and poles) is used to classify the motion of a point cloud [14–16].

The advantages of occupancy grid-based static point cloud classification are that its implementation is relatively straightforward and its performance is stable because it has been studied for a long time in various applications. However, occupancy grid mapping has several disadvantages for use in real-time automated and intelligent vehicle applications. Large memory is required because the driving environment must be represented by a grid or voxel cells. Also, the ray tracing method takes a long time to update all cells related to all LiDAR beams. In addition, because space is represented by discrete cells, a discretization error occurs when the resolution is coarse.

Research on the detection and tracking of moving objects using LiDAR has been conducted to recognize the driving environment of automated and intelligent vehicles [17,18]. Object detection algorithms detect surrounding objects by clustering the point cloud and generate a bounding box for the each detected object. Tracking algorithms generate tracks for detected objects to estimate their position, direction, velocity, and acceleration. Using the tracking results, we can classify the motion of a point cloud into dynamic and static states. Points in a track bounding box above a certain speed are classified as dynamic, and the remaining points are classified as static. This tracking-based point motion classification is straightforward, and the tracking results can be reused. However, it has some limitations. The point cloud clustering groups the detected points on the same object in the object detection step. Although many clustering methods have been studied, it is difficult to obtain accurate results using point cloud information alone. Incorrect clustering causes incorrect point motion classification. In addition, because the tracking has an initialization time to generate new track, it struggles to satisfy real-time motion classification. Furthermore, because it is difficult for tracking to accurately estimate the speed of slow objects, the point motion of objects is likely to be misclassified near the threshold speed.

The point motion classification algorithm presented in this paper has many advantages over previously proposed ones. First, the proposed method directly segments the point cloud into dynamic and static states using the laser beam model. Therefore, there is no chance of misclassification due to discretization error in the occupancy grid method and erroneous clustering of the tracking-based method. In addition, the proposed method does not require a large amount of memory like the occupied grid approach because it simply buffers recent point clouds. Finally, the proposed method is able to satisfy the real-time requirements of point motion classification because it does not need to update all of the gird cells to initialize a new track.

3. System Architecture

The objective of the proposed point motion segmentation algorithm is to classify the latest LiDAR point, $z_t^{n,m}$, into motion states, $motion_t^{n,m}$, in real-time. The notation of the LiDAR point and motion states are $z_{time}^{index,order}$ and $motion_{time}^{index,order}$, respectively. The n describes the index of the laser beam from 1 to N. The m represents the order of multi-echo for the laser beam and usually has a value of 2 or less. The t represents the time for the LiDAR scan measurement. The "motion state" has three possible values: $motion = \{dynamic, static, unknown\}$. The *dynamic* state indicates the points detected as moving objects, and the *static* state represents points detected as stationary objects. The *unknown* state means that there is not sufficient evidence to classify the motion state as *dynamic* or *static*.

The basic principle of the proposed algorithm is to classify the motion state of the current point cloud by applying the laser characteristic model to the registration relationship between the previously measured point cloud and the current point cloud. The inputs of the proposed algorithm are the current point cloud, $Z_t = \{z_t^{1,m}, z_t^{2,m}, \cdots, z_t^{N,m}\}$; the previously buffered point clouds, $Z_{t-1}, \cdots, Z_{t-W+1}, Z_{t-W}$; and the sensor pose, $x_t, x_{t-1}, \cdots, x_{t-W+1}, x_{t-W}$, for each point cloud. W denotes the time window size of the previous data buffer to be used for motion classification of the current point cloud. There are several methods for obtaining the sensor pose $x_t, \cdots, x_{t-W}$, such as inertial measurement unit (IMU) dead reckoning, scan matching, a high-definition (HD) map-based localization, and simultaneous localization and mapping (SLAM). To avoid loss of generality, we assume that the sensor's pose and its uncertainty are provided. The output of the algorithm is the motion state $Motion_t = \{motion_t^{1,m}, motion_t^{2,m}, \cdots, motion_t^{N,m}\}$ of $Z_t = \{z_t^{1,m}, z_t^{2,m}, \cdots, z_t^{N,m}\}$.

The point motion segmentation algorithm consists of two steps: (1) probabilistic modeling of point motion and (2) evidential point motion classification. In the first step, the probability of $Motion_{(t-k)\to t}$, which is a motion classification of Z_t against Z_{t-k}, is updated. Figure 2 illustrates the concept of the probability update of $Motion_{(t-k)\to t}$ based on a geometrical relationship between (Z_t, x_t) and (Z_{t-k}, x_{t-k}). The likelihood field of the motion can be updated using (Z_{t-k}, x_{t-k}) and the characteristics of the laser (such as beam divergence and multi-echo). The *Dynamic* probability for $z_t^{m,1}$ and $z_t^{m+3,1}$ will be higher when they are located in the path of the laser (green region) for the previous point cloud Z_{t-k}, and the *Static* probability for $z_t^{m+2,1}$ will be higher if it is located near the previous point (red region).

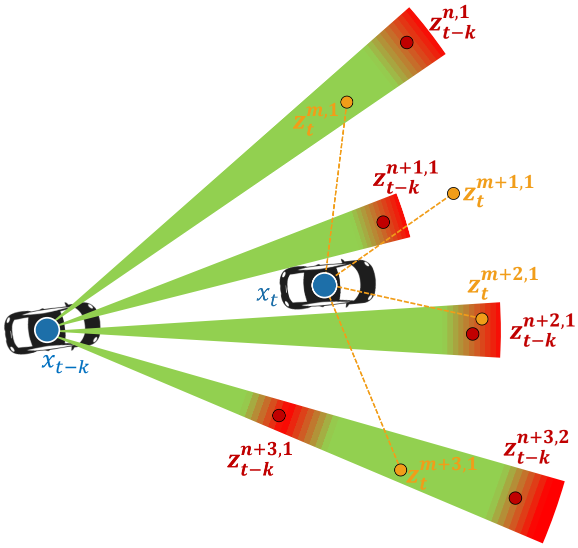


Figure 2. Probability update of $Motion_{(t-k)\to t}$. The motion probability of the point cloud Z_t can be updated based on the geometric relationship for the previous point cloud Z_{t-k}.

In the second step, the probabilities of each motion classification $Motion_{(t-1)\to t}, \cdots, Motion_{(t-W)\to t}$ are integrated to estimate the final motion classification $Motion_t$, as shown in Figure 3. However, the probability of $Motion_{(t-k)\to t}$ cannot be updated by previous points if the current points are not in the likelihood field of the previous points. In this case, it should be classified as *unknown*. However, because *unknown* cannot be expressed clearly using probability theory, evidence theory, which can handle the *unknown* state explicitly, is employed. The probabilities of $Motion_{(t-1)\to t}, \cdots, Motion_{(t-W)\to t}$ are converted into mass (degree of belief) with consideration of LiDAR and sensor pose uncertainty and then integrated into a mass of $Motion_t$ using Dempster's combination rule. The motion states $Motion_t = \{motion_t^{1,m}, motion_t^{2,m}, \cdots, motion_t^{N,m}\}$ of each point $Z_t = \{z_t^{1,m}, z_t^{2,m}, \cdots, z_t^{N,m}\}$ are determined using the integrated mass of $Motion_t$.

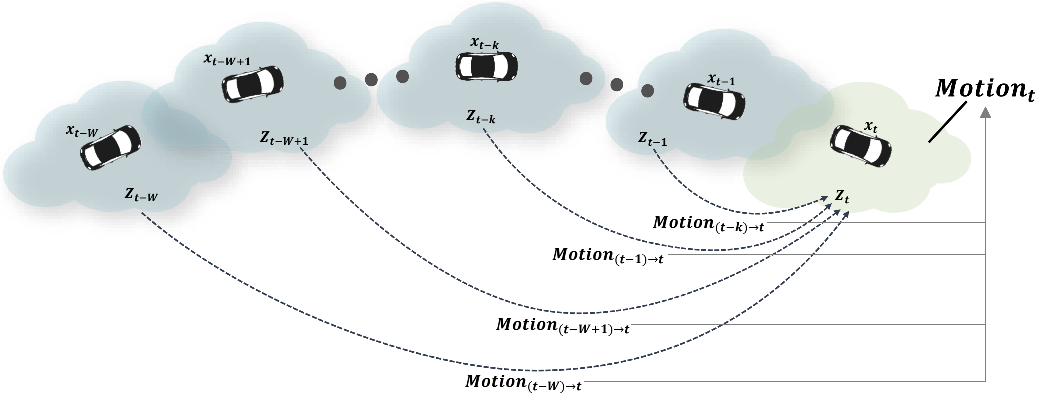


Figure 3. Motion probabilities $p(Motion_{(t-1)\to t}), \cdots, p(Motion_{(t-W)\to t})$ are integrated into the latest motion probability $p(Motion_t)$.

4. Probabilistic Modeling of LiDAR Point Motion

4.1. Characteristics of LiDAR Point Cloud

LiDAR uses rotating laser beams to measure the distances and angles from surrounding objects. A laser pulse is emitted at a specific angle, and the distance to the object for that angle can be measured using the time-of-flight (ToF) principle, as demonstrated in Figure 4. ToF represents the difference between the time the laser pulse is emitted from the diode and the time it returns to the object. The distance is calculated by multiplying this time by the speed of the laser light. Using the horizontal–vertical emitted angles and the measured distances, 3D information of surrounding objects can be reconstructed in the form of point data.

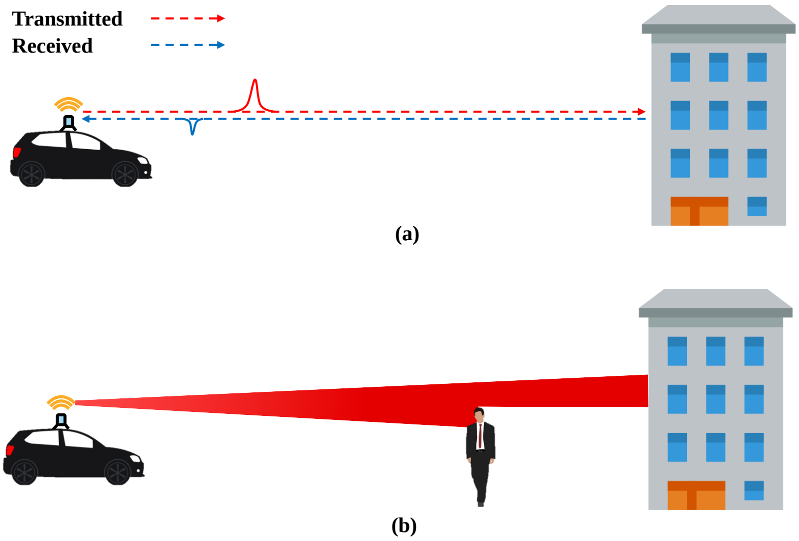


Figure 4. Light detection and ranging (LiDAR) measurement characteristics: (**a**) Time-of-flight (ToF) and (**b**) laser beam divergence and multi-echo.

The actual LiDAR's laser is not emitted in a straight line, as shown in the Figure 4a. The laser has a characteristic of "beam divergence", which increases the beam's cross section over the distance. Because of this characteristic, the farther away from the laser source an object is, the wider the area objects can be detected in. In addition, beam divergence enables multi-echoing of the emitted laser pulse, as shown in Figure 4b, and the multi-echo allows simultaneous measurement of distances to various objects. Another important characteristic of LiDAR is the uncertainty of the distance and angular measurement. Despite the use of lasers, the distance measurement is not infinitely accurate. The distance measurement accuracy is proportional to the measuring capability of the time the laser pulse takes to return. Conversely, the angular accuracy (vertical and horizontal) is discretely accurate because LiDAR is able to control and configure the emission angle. Many previous studies that used LiDAR measurements did not properly account for the above-mentioned characteristics of LiDAR (i.e., beam divergence and distance uncertainty); they treated LiDAR measurements as points with no volume and constant 3D Gaussian uncertainty. To classify LiDAR point motion with high accuracy and reliability, the proposed algorithm accurately reflects these characteristics of LiDAR.

The LiDAR point cloud measurement $Z_t = \{z_t^{1,m}, z_t^{2,m}, \cdots, z_t^{N,m}\}$ must be representable in both spherical and Cartesian coordinates for processing by the point motion segmentation algorithm. The measurement can be represented in spherical coordinates as $Z_{r\theta\phi,t} = \{z_{r\theta\phi,t}^{1,m}, \cdots, z_{r\theta\phi,t}^{i,m}, \cdots, z_{r\theta\phi,t}^{N,m}\}$. The point $z_{r\theta\phi,t}^{i,m}$ is represented as $z_{r\theta\phi,t}^{i,m} = \{r_t^{i,1}, \cdots, r_t^{i,m}, \theta_t^i, \phi_t^i\}$, where r is distance to a point with a second echo, θ is the vertical (polar) angle of the point, ϕ is the horizontal (azimuthal) angle of the point, and m is the number of echos. The Cartesian coordinate representation is $Z_{xyz,t} = \{z_{xyz,t}^{1,m}, \cdots, z_{xyz,t}^{i,m}, \cdots, z_{xyz,t}^{N,m}\}$, where $z_{xyz,t}^{i,m} = \{x_t^{i,m}, y_t^{i,m}, z_t^{i,m}\}$.

4.2. Probabilistic Modeling for LiDAR Point Motion

The point motion segmentation classifies the latest N LiDAR points $z_t^{1,m}, z_t^{2,m}, \cdots, z_t^{N,m}$ into the motion states $motion_t^{1,m}, motion_t^{2,m}, \cdots, motion_t^{N,m}$, respectively, in real-time, where $motion$ consists of three states $\{dynamic, static, unknown\}$. The motion state of the latest point cloud is segmented based on the registration relationship for the previously buffered point cloud. In other words, the latest point cloud $Z_t = \{z_t^{1,m}, z_t^{2,m}, \cdots, z_t^{N,m}\}$ is segmented into $Motion_t = \{motion_t^{1,m}, motion_t^{2,m}, \cdots, motion_t^{N,m}\}$ based on the W-buffered point clouds $Z_{t-1}, \cdots, Z_{t-W+1}, Z_{t-W}$ and the sensor pose $x_t, x_{t-1}, \cdots, x_{t-W+1}, x_{t-W}$ of each point cloud. The sensor pose $x_t, \cdots, x_{t-W}$ can be obtained using several methods, such as an IMU dead reckoning, scan matching, HD map-based localization, and SLAM. However, the proposed algorithm assumes that the sensor's pose information and its uncertainty is abstracted regardless of the type of pose estimation method.

Probabilistic motion modeling of the point cloud Z_t can be estimated using the LiDAR sensor pose x_t of Z_t and the previously detected LiDAR point cloud Z_{t-k} and its sensor pose x_{t-k}, as shown in Figure 2. The probabilistic motion model of Z_t to x_t, Z_{t-k}, and x_{t-k} can be described as $p(Motion_{(t-k)\to t})$. The probability $p(Motion_{(t-k)\to t})$ can be represented by a conditional probability for the given conditions, the past and present point cloud pairs (Z_t, Z_{t-k}), and their sensor pose (x_t, x_{t-k}), as represented in Equation (1).

$$p(Motion_{(t-k)\to t}) = p(Motion_t | Z_t, Z_{t-k}, x_t, x_{t-k}) \tag{1}$$

$Motion_t$ is composed of each independent point motion $\{motion_t^{1,m}, motion_t^{2,m}, \cdots, motion_t^{N,m}\}$, so $p(Motion_{(t-k)\to t})$ can be represented by the set of conditional probabilities of each point, as described by

$$p(Motion_{(t-k)\to t}) = \{p(motion_t^{1,m} | Z_t, Z_{t-k}, x_t, x_{t-k}), \ldots, p(motion_t^{N,m} | Z_t, Z_{t-k}, x_t, x_{t-k})\}. \tag{2}$$

$motion_t^{i,m}$ consists of two states $\{dynamic, static\}$, and the sum of $p(dynamic)$ and $p(static)$ is always one.

The conditional probability of one point motion can be reorganized by the Bayes rule, as represented by Equation (3).

$$p(motion_t^{i,m}|z_t^{i,m}, Z_{t-k}, x_t, x_{t-k}) = \frac{p(z_t^{i,m}|motion_t^{i,m}, Z_{t-k}, x_t, x_{t-k})p(motion_t^{i,m}|Z_{t-k}, x_t, x_{t-k})}{p(z_t^{i,m}|Z_{t-k}, x_t, x_{t-k})} \quad (3)$$

$p(z_t^{i,m}|motion_t^{i,m}, Z_{t-k}, x_t, x_{t-k})$ is the likelihood of the LiDAR point measurement for the given motion state. $p(motion_t^{i,m}|Z_{t-k}, x_t, x_{t-k})$ is the predicted probability density function. The motion for given Z_{t-k}, x_t, and x_{t-k} can be represented by a uniform distribution, so $p(motion_t^{i,m} = \{\text{static or dynamic}\}|Z_{t-k}, x_t, x_{t-k})$ is 0.5. $p(z_t^{i,m}|Z_{t-k}, x_t, x_{t-k})$ can be a normalization factor by applying marginalization, as described by Equation (4).

$$p(motion_t^{i,m}|z_t^{i,m}, Z_{t-k}, x_t, x_{t-k}) = \sum_{motion} p(z_t^{i,m}|motion_t^{i,m}, Z_{t-k}, x_t, x_{t-k})p(motion_t^{i,m}|Z_{t-k}, x_t, x_{t-k}) \quad (4)$$

By summarizing the above equations, the posterior probability of one point motion can be represented by the following equation,

$$p(motion_t^{i,m}|z_t^{i,m}, Z_{t-k}, x_t, x_{t-k}) = \eta \; p(z_t^{i,m}|motion_t^{i,m}, Z_{t-k}, x_t, x_{t-k}) \quad (5)$$

where η is the normalization factor $p(motion_t^{i,m}|z_t^{i,m}, Z_{t-k}, x_t, x_{t-k})$ of Equation (4). Equation (5) is the conditional probability of one LiDAR point motion being expressed by the likelihood of the given point motion. Therefore, the motion probability estimation problem is converted to a likelihood estimation problem for LiDAR point cloud.

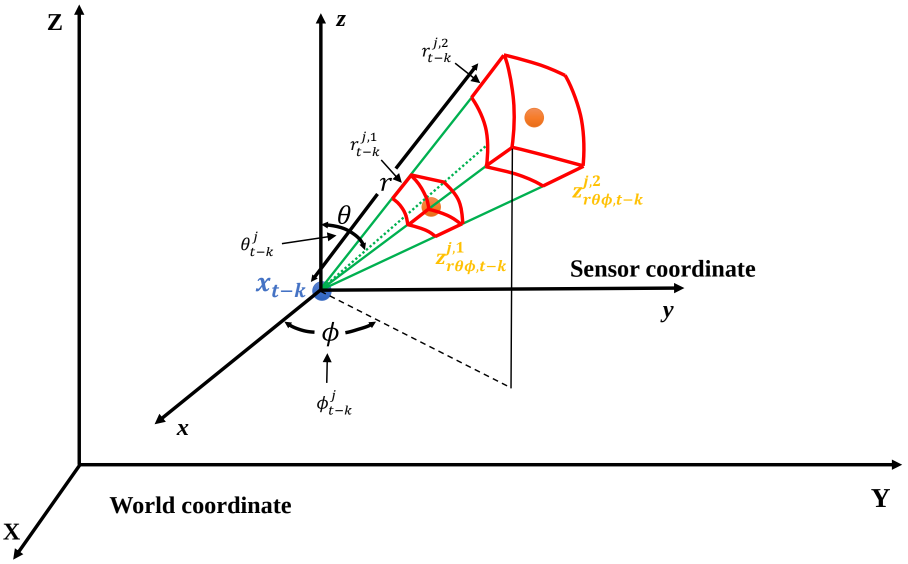


Figure 5. Likelihood field for the point $z_t^{i,m}$ is constructed by previous point $z_{r\theta\phi,t-k}^{j,l}$ with consideration of the beam divergence.

4.3. Likelihood of LiDAR Point Measurement

We know that the point cloud motion probability $p(Motion_{(t-k)\to t})$ can be obtained from the likelihood of the point cloud $p(z_t^{i,m}|motion_t^{i,m}, Z_{t-k}, x_t, x_{t-k})$, as described by Equation (5). The likelihood $p(z_t^{i,m}|motion_t^{i,m}, Z_{t-k}, x_t, x_{t-k})$ represents a statistical state when $motion_t^{i,m}$ is determined as *static* or *dynamic* for given Z_{t-k}, x_t, and x_{t-k}. The likelihood field of one point $z_t^{i,m}$ can be represented intuitively, as shown in Figure 5. The likelihood field of the LiDAR point $z_t^{i,m}$, measured at sensor pose x_t at time t, is represented by points $z_{t-k}^{j,l}$ in point cloud Z_{t-k} measured at pose x_{t-k} at the previous time $t-k$. The intensity of the green color indicates the likelihood of the point $z_t^{i,m}$ in the green region being in dynamic motion, and the intensity of the red region represents the likelihood of the point $z_t^{i,m}$ in the red region being in static motion. The likelihood field is represented in the local spherical coordinates of the previous sensor pose x_{t-k}. The point cloud Z_{t-k} is represented in spherical coordinates as $Z_{t-k} = Z_{r\theta\phi,t-k} = \{z_{r\theta\phi,t-k}^{1,l}, \cdots, z_{r\theta\phi,t-k}^{j,l}, \cdots, z_{r\theta\phi,t-k}^{J,l}\}$. The likelihood field for the point $z_t^{i,m}$ is constructed in a triangular-pyramid form by each previous point $z_{r\theta\phi,t-k}^{j,l}$ with consideration of the beam divergence characteristics of the LiDAR laser, as shown in Figure 5.

The 3D likelihood distribution of $p(z_t^{i,m}|motion_t^{i,m}, Z_{t-k}, x_t, x_{t-k})$ can be divided by two 2D likelihood fields. The first one is a likelihood field for the distance–horizontal angle $(r-\phi)$ plane, $p(r_t^{i,m}, \phi_t^i|motion_t^{i,m}, Z_{t-k}, x_t, x_{t-k})$, and the second is a likelihood field of the distance-vertical angle $(r-\theta)$ plane, $p(r_t^{i,m}, \theta_t^{i,m}|motion_t^i, Z_{t-k}, x_t, x_{t-k})$. Figure 6 shows the 2D likelihood field in the $(r-\phi)$ and $(r-\theta)$ planes for the previous measurements $z_{t-k}^{j,1}, z_{t-k}^{j,2}, z_{t-k}^{j+1,1}$ and $z_{t-k}^{j+2,1}$. For each horizontal and vertical angle, the likelihood field is distributed discretely based on the resolution of the horizontal and vertical laser pulses. Due to the beam divergence of the laser, the likelihood fields gradually disperse.

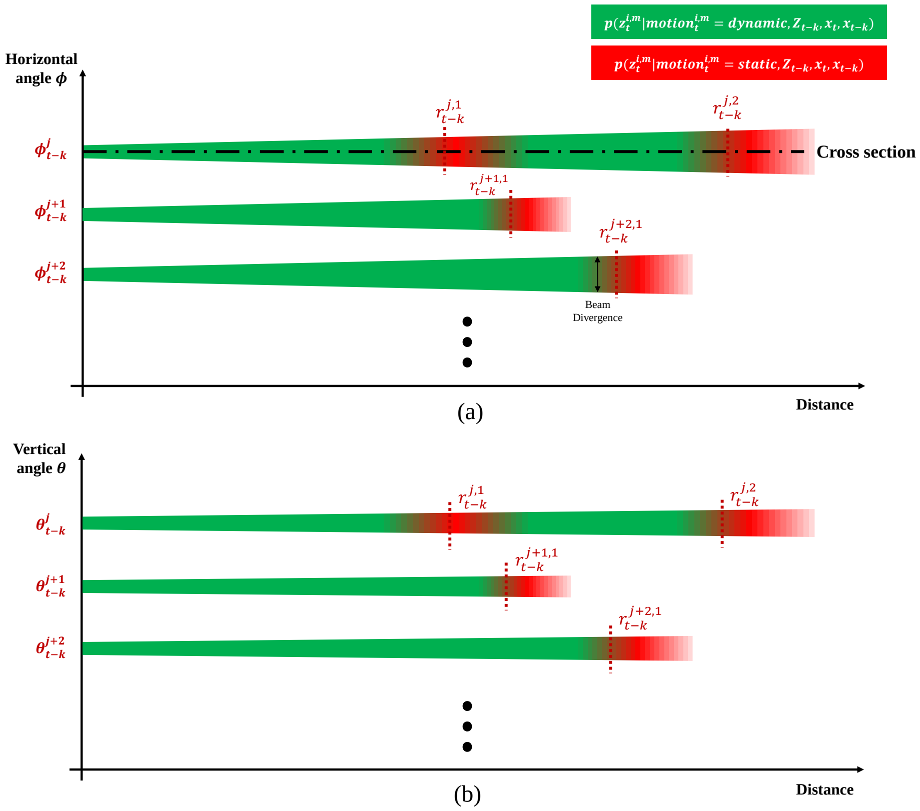


Figure 6. 2D likelihood field for the distance (r) and angle (horizontal ϕ (**a**) and vertical θ (**b**)) of the previous measurements $z_{t-k}^{j,1}, z_{t-k}^{j,2}, z_{t-k}^{j+1,1}$ and $z_{t-k}^{j+2,1}$.

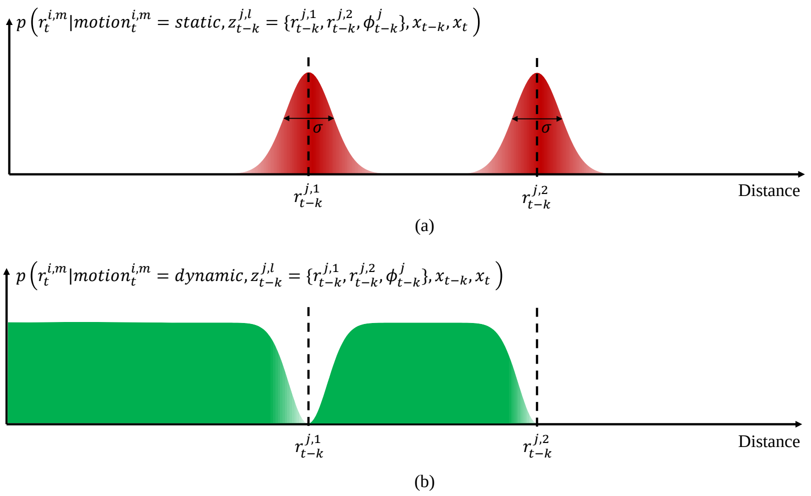


Figure 7. Likelihood of *static* (**a**) and *dynamic* (**b**) for one laser beam.

The cross-section of the likelihood field for one laser beam $z_{t-k}^{j,l}$ in Figure 6a can be represented by the likelihood value in Figure 7. Through this figure, we can more accurately analyze the distribution of likelihood for each motion. Figure 7a shows the likelihood $p(z_t^{i,m} | motion_t^{i,m} = static, z_{t-k}^{j,l}, x_t, x_{t-k})$ when a measured point $z_t^{i,m}$ is *static* for given previous measurement $z_{t-k}^{j,l}$ and given poses x_t and x_{t-k}. Here, the previous measurement $z_{t-k}^{j,l}$ can be expressed in spherical coordinates as $\{r_{t-k}^{j,1}, r_{t-k}^{j,2}, \phi_{t-k}^{j}\}$. The region where the previous LiDAR point was detected is likely to be static. LiDAR is measured using ToF, so the uncertainty of the measured distance, $r_t^{i,m}$, depends on the accuracy of the ToF sensor. Considering this uncertainty, the likelihood of static motion can be expressed as a Gaussian distribution, as described by Figure 7a and Equation (6).

$$p(r_t^{i,m} | motion_t^{i,m} = static, z_{t-k}^{j,l}, x_t, x_{t-k}) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(r - r_t^{i,m})^2/2\sigma^2} \tag{6}$$

σ is the standard deviation of the distance measurement, which are different depending on the LiDAR. Figure 7b shows the likelihood $p(z_t^{i,m} | motion_t^{i,m} = dynamic, z_{t-k}^{j,l}, x_t, x_{t-k})$ when a measured point $z_t^{i,m}$ is *dynamic* for given $z_{t-k}^{j,l}, x_t$, and x_{t-k}. The region where the previous LiDAR beam $z_{t-k}^{j,l}$ passed is likely to be free, and the location of the current LiDAR point $z_t^{i,m}$ in the region means that this point is more likely to be detected from a dynamic object. This characteristic can be represented by the following equation:

$$p(r_t^{i,m} | motion_t^{i,m} = dynamic, z_{t-k}^{j,l}, x_t, x_{t-k}) = \\ \begin{cases} MaxLikelihood_{t-k}^{i,m} - p(r_t^{i,m} | motion_t^{i,m} = static, z_{t-k}^{j,l}, x_t, x_{t-k}) & r <= r_{t-k}^{j,max(m)} \\ 0 & else \end{cases} \tag{7}$$

Here, $MaxLikelihood_{t-k}^{i,m}$ denotes the maximum likelihood value for the previous point measurement $z_{t-k}^{j,l}$ and can be represented by the following equation:

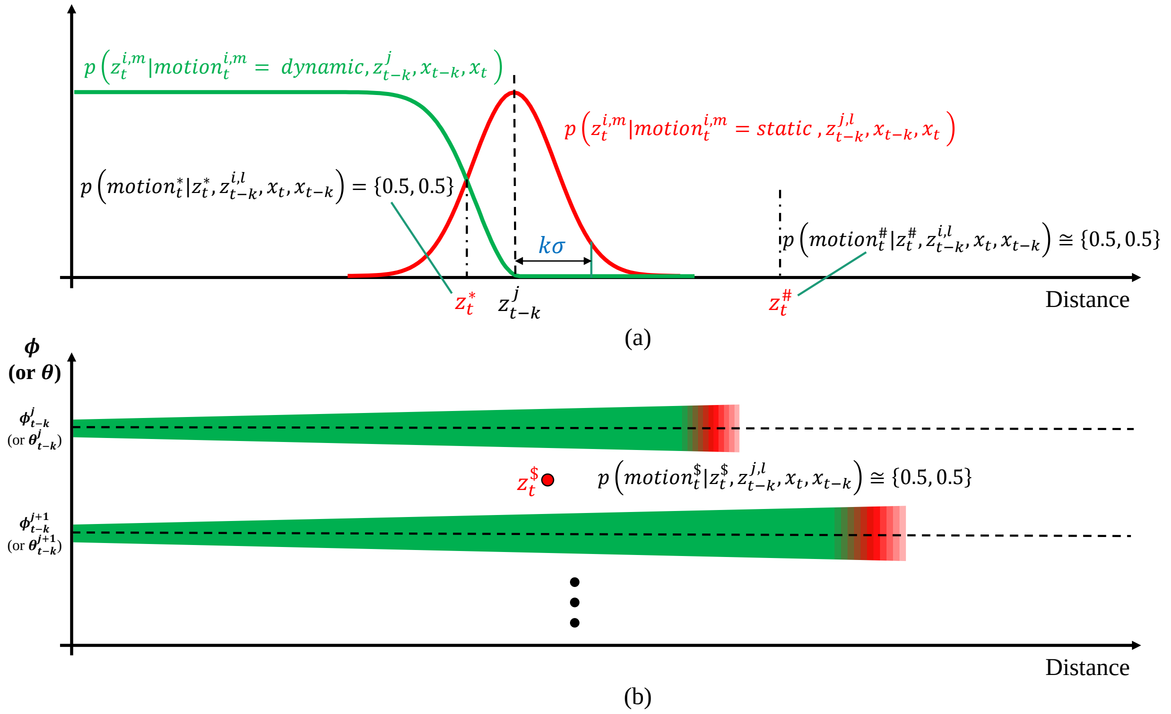
$$MaxLikelihood_{t-k}^{i,m} = p(r_{t-k}^{j,l} | motion_t^{i,m} = static, z_{t-k}^{j,l}, x_t, x_{t-k}) \tag{8}$$

When the point $z_t^{i,m}$ is located in the likelihood field for the given $\{z_{t-k}^{j,l}, x_{t-k}, x_t\}$, we can obtain the likelihood through Equations (6) and (7). Then, the probability of the point motion can be calculated through Equation (5). However, if the point $z_t^{i,m}$ is located outside of the likelihood field, we cannot obtain the likelihood using the above equations. The area outside of the likelihood field must be dealt with as *unknown*, but probability theory cannot handle the *unknown* state explicitly. Therefore, in the next chapter, we apply evidence theory to deal with the *unknown* state explicitly.

5. Evidential Point Motion Classification

5.1. Evidential Modeling of LiDAR Point Motion

The process of point-wise probabilistic motion estimation achieved through the likelihood field was described in the previous section. However, there is a limitation to the probabilistic method if the point $z_t^{i,m}$ is not located above the likelihood field. We can see this limitation in Figure 8. In Figure 8a, the measurement z_t^* represents a normal case because the point is located inside the likelihood field. The point motion probability $p(motion_t^*|z_t^*, z_{t-k}^i, x_t, x_{t-k})$ can be $\{static, dynamic\} = \{0.5, 0.5\}$ because the likelihoods for *dynamic* and *static* are the same. However, the probabilities of motion for $z_t^\#$ in Figure 8a and $z_t^\$$ in Figure 8b are not included in the likelihood field close to $\{0.5, 0.5\}$, because both likelihoods for *dynamic* and *static* are zero. This means that we cannot distinguish the difference between z_t^*, $z_t^\#$, and $z_t^\$$ in the probabilistic approach. To overcome this limitation, an evidential approach (Dempster–Shafer theory) is applied to explicitly distinguish the motion of points that are not located in the likelihood field.



$p\left(z_t^{i,m}|motion_t^{i,m} = dynamic, z_{t-k}^j, x_{t-k}, x_t\right)$
$p\left(z_t^{i,m}|motion_t^{i,m} = static, z_{t-k}^{j,l}, x_{t-k}, x_t\right)$
$p\left(motion_t^*|z_t^*, z_{t-k}^{i,l}, x_t, x_{t-k}\right) = \{0.5, 0.5\}$
$k\sigma$
$p\left(motion_t^\#|z_t^\#, z_{t-k}^{i,l}, x_t, x_{t-k}\right) \cong \{0.5, 0.5\}$
z_t^*
z_{t-k}^j
$z_t^\#$
Distance
(a)
ϕ (or θ)
ϕ_{t-k}^j (or θ_{t-k}^j)
$z_t^\$$
$p\left(motion_t^\$|z_t^\$, z_{t-k}^{j,l}, x_t, x_{t-k}\right) \cong \{0.5, 0.5\}$
ϕ_{t-k}^{j+1} (or θ_{t-k}^{j+1})
Distance
(b)

Figure 8. Problems of probabilistic motion segmentation. The probabilities of motion for $z_t^\#$ (**a**) and $z_t^\$$ (**b**) are close to $\{0.5, 0.5\}$, because both likelihoods for *dynamic* and *static* are zero.

Both probabilistic and evidential approaches are based on the concept of assigning weights to the hypothesized states of the measurement. However, the evidential approach allows sets of alternatives, which means new states can be created by combining existing states. The probabilistic approach deals with the two states $\{static, dynamic\}$. In the evidential approach, the two states form a frame of discernment $\Omega = \{static, dynamic\}$. Dempster–Shafer theory can manage more states explicitly (Ω, ϕ) by extending the frame of discernment Ω to the power set $2^\Omega = \{static, dynamic, \Omega, \phi\}$. Ω is the set $\Omega = \{static, dynamic\}$, which means that the point motion is *static* or *dynamic*. However, because the point motion cannot be *static* and *dynamic* simultaneously, the state Ω indicates an *unknown* state. ϕ is an empty set, which means that the point motion is not both *static* and *dynamic*. However, because this situation is physically impossible, the state ϕ indicates a *conflict* situation. For each state of the power set $2^{\Omega} = \{static, dynamic, unknown, conflict\}$ in the evidential approach, a mass function *Mass* is used to quantify the belief of the hypothesis. The mass functions of $Mass_t^{i,m}(static)$ and $Mass_t^{i,m}(dynamic)$ represent the belief of point $z_t^{i,m}$ being *static* and *dynamic*, respectively. The mass function of $Mass_t^{i,m}(unknown)$ is the union of the beliefs of *static* and *dynamic*, and $Mass_t^{i,m}(conflict)$ represents the belief that the point is conflicted by different measurements. The sum of mass functions for the power set must be *one* based on its definition in the evidential framework.

Based on the evidential approach, we can explicitly handle the points located outside the likelihood field for the given point $z_{t-k}^{j,l}$ as an *unknown* state. The boundary between the inside and outside is $k\sigma$, as shown in Figure 8. k is the tuning factor, which determines the size of the likelihood boundary, and we used $k = 3$. The point z_t^* is located inside the likelihood field, but $z_t^{\#}$ and $z_t^{\$}$ are located outside the likelihood field. The *mass* of point $z_t^{i,m}$ motion for the given $z_{t-k}^{j,l}, x_t$, and x_{t-k} is denoted by $mass_{(t-k)\to t}^{j,l\to i,m}(state)$ for each $state = \{static, dynamic, unknown, conflict\}$. $mass_{(t-k)\to t}^{j,l\to i,m}(state)$ can be calculated based on whether the point $z_t^{i,m}$ is located inside or outside of the likelihood field using the following equation.

$$\begin{aligned}
mass_{(t-k)\to t}^{j,l\to i,m}(static) &= \begin{cases} \lambda_k p(motion_t^{i,m} = static | z_t^{i,m}, z_{t-k}^{j,l}, x_t, x_{t-k}), & \text{inside} \\ 0, & \text{outside} \end{cases} \\
mass_{(t-k)\to t}^{j,l\to i,m}(dynamic) &= \begin{cases} \lambda_k p(motion_t^{i,m} = dynamic | z_t^{i,m}, z_{t-k}^{j,l}, x_t, x_{t-k}), & \text{inside} \\ 0, & \text{outside} \end{cases} \\
mass_{(t-k)\to t}^{j,l\to i,m}(unknown) &= \begin{cases} 1 - Mass_{(t-k)\to t}^{j,l\to i,m}(static) - Mass_{(t-k)\to t}^{j,l\to i,m}(dynamic), & \text{inside} \\ 1, & \text{outside} \end{cases} \\
mass_{(t-k)\to t}^{j,l\to i,m}(conflict) &= 0
\end{aligned} \tag{9}$$

The *mass* values of the *static* and *dynamic* states are calculated by the motion probability $p(motion_t^{i,m} | z_t^{i,m}, z_{t-k}^{j,l}, x_t, x_{t-k})$ and its confidence, λ_k. The confidence λ_k can be determined using Equation (10).

$$\lambda_k = \lambda_{reg} exp(-\frac{k}{\tau}) \tag{10}$$

λ_{reg} describes the confidence of the pose registration between x_t and x_{t-1}. This value is determined by the performance of the registration method, such as IMU, scan matching, and HD mapping. If the registration is very accurate, the value is close to one; however, if it is not good, it is close to zero. The confidence λ_k is also affected by the time difference k. Because the confidence of the probabilistic model decreases as the time difference k increases, the confidence λ_k also decreases by $exp(-k/\tau)$, where τ is the time constant that determines the decay rate.

5.2. Point Motion Segmentation by Integrating the Point Motion Masses

For the given point $z_{t-k}^{j,l}$ and the given poses x_t and x_{t-k}, the point motion of $z_t^{i,m}$ can be described by mass function $mass_{(t-k)\to t}^{j,l\to i,m}(state)$. For all given previous scan points $Z_{t-k} = \{z_{t-k}^{1,l}, \cdots, z_{t-k}^{j,l}, \cdots, z_{t-k}^{N,l}\}$ and the given poses x_t and x_{t-k}, several mass functions $mass_{(t-k)\to t}^{1,l\to i,m}(state) \cdots mass_{(t-k)\to t}^{N,l\to i,m}(state)$ can be calculated. We must integrate the mass functions into one mass function $mass_{(t-k)\to t}^{i,m}(state)$. In addition, for the previously buffered point clouds $Z_{t-1}, \cdots, Z_{t-W+1}, Z_{t-W}$ in the time window W, several mass functions $Mass_{(t-1)\to t}^{i,m}(state), \cdots, Mass_{(t-W)\to t}^{i,m}(state)$ can be obtained, and these mass functions should be integrated into a single mass function $Mass_t^{i,m}(state)$ to represent the motion of one point $z_t^{i,m}$. To integrate two different mass values from different laser scans and times, Dempster's combination rule (Equation (11)) is applied.

$$Mass_1 \oplus Mass_2 \Longrightarrow \tag{11}$$

$$Mass_{1\oplus 2}(A) = \frac{Mass_{1\cap 2}(A)}{1 - Mass_{1\cap 2}(\phi)}, Mass_{1\oplus 2}(\phi) = 0,$$
$$\forall A \subseteq \Omega, A \neq \phi \tag{12}$$

Dempster's combination rule is based on the conjunctive combination rule described by Equation (13).

$$Mass_{1\cap 2}(A) = \sum_{B\cap C = A | B, C \subseteq \Omega} Mass_1(B) \cdot Mass_2(B) \tag{13}$$

6. Experiments

6.1. Experimental Environments

An autonomous vehicle (A1) was used for the experiment to evaluate the proposed algorithm. A1 was equipped with two LiDARs (Velodyne VLP-16) and an IMU, as shown in Figure 9. The LiDARs provided point cloud data with a 10 Hz sampling frequency and their maximum detection range is 100 m.

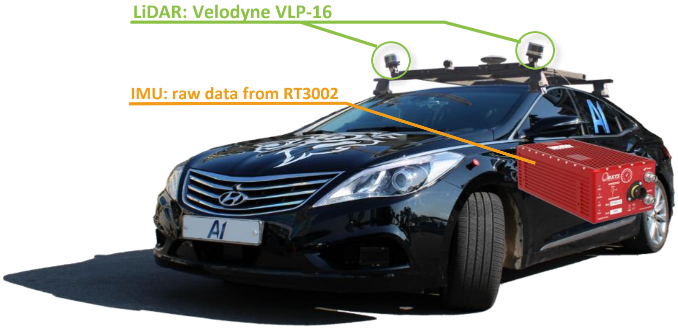
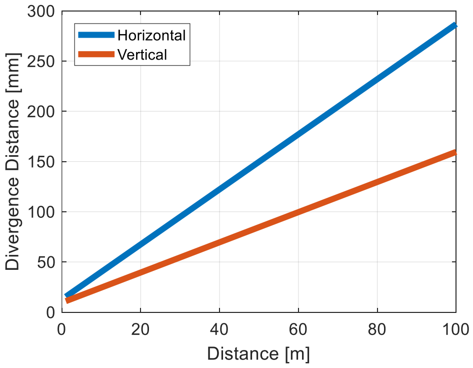


Figure 9. Test vehicle and sensor (LiDAR and positioning system) configuration.

LiDAR lasers beams have beam divergence, which means the beam cross section is increased over time. We designed a beam divergence model based on the specification of the LiDAR sensors, as shown in Figure 10. The horizontal and vertical beam divergence characteristics were different. The standard deviation σ of the distance accuracy was set to 3 cm. Because the distance accuracy can vary based on factors such as temperature and target reflectivity, the selection of the standard deviation σ for the probabilistic LiDAR model must consider the uncertainty. The horizontal field of view (FoV) was 360°, and the horizontal angular resolution was set to 0.2°. The vertical FoV was 30°, and the vertical resolution was 2°. Because the LiDAR controls the laser emitting angle, it was assumed that the angular uncertainties (vertical and horizontal) were negligible.

Distance (mm)	Horizontal (mm)	Vertical (mm)
1000	15.4	11.0
2000	18.2	12.5
5000	26.4	17.0
10000	40.1	24.5
15000	53.8	32.0
25000	81.2	47.0
50000	149.7	84.6
100000	286.8	159.6

(a)



(b)

Figure 10. LiDAR beam divergence model: (**a**) divergence characteristics from VLP-16 specification and (**b**) divergence length over detection distance.

Dead reckoning was implemented using the IMU to estimate the LiDAR pose for each time step. For the experiments, raw IMU data (acceleration and gyro) from the RT3002 sensor were used without real-time kinematic (RTK) GNSS correction. The specifications of the MEMS IMU are listed in Table 1. Although the IMU was not sufficiently accurate to estimate the long-term pose of the LiDAR sensor, it can provide stable performance in short windows of approximately 50 or less (five seconds or less). In addition, the evidential integration algorithm was able to account for the inaccuracy of the IMU-based pose estimation by tuning the registration confidence λ_{reg}. By considering the installed MEMS IMU, we set λ_{reg} to 0.9.

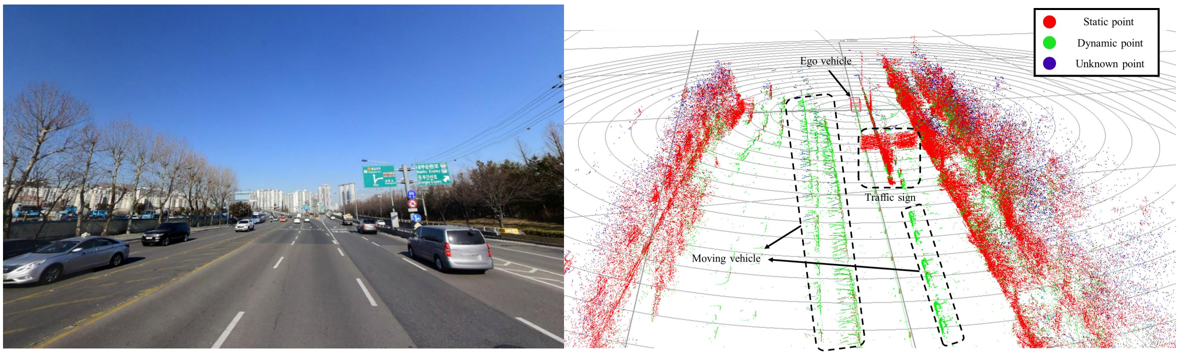
Table 1. Specification of IMU sensor. An IMU was used inside the GNSS/INS (RT3002) to estimate the pose of the LiDAR sensor.

IMU		Performance
Acceleration	**Bias**	$10 mm/s^2$ (1σ)
	Linearity	0.01%
	Scale Factor	0.1% (1σ)
	Range	$100 m/s^2$
Angular	**Bias**	$0.01°/s$ (1σ)
	Scale Factor	0.1% (1σ)
	Range	$100°/s$

The synchronization between the LiDAR, IMU, and point motion classification algorithm was measured by a pulse per second (PPS) signal from an RT3002. The LiDARs and IMU were precisely calibrated to be located in the same coordinate system.

6.2. Segmentation Performance Evaluation through Comparative Analysis

To evaluate the performance of the point-wise motion segmentation, experiments were conducted under various scenarios (e.g., cities and highways). The total length of the experiment road is more than two kilometers. Figure 11a shows single scene of the experimental condition, where moving cars and stationary road structures were mixed. The result of segmentation through the proposed algorithm is shown in Figure 11b. The RGB value for each point is set using the proposed motion belief algorithm. The red values represent *static* state belief, the green values represent *dynamic* state belief, and the blue values represent *unknown* state belief. Therefore, the objects that have a high probability of stopping will appear red, moving objects will appear green, and unsegmented objects will appear blue. As shown in Figure 11b, traffic signs and roadside trees are segmented as red, and moving cars are classified as green.



(a) (b)

Figure 11. (**a**) Test site with moving cars and stationary road structures and (**b**) classification results of point-wise LiDAR motion segmentation.

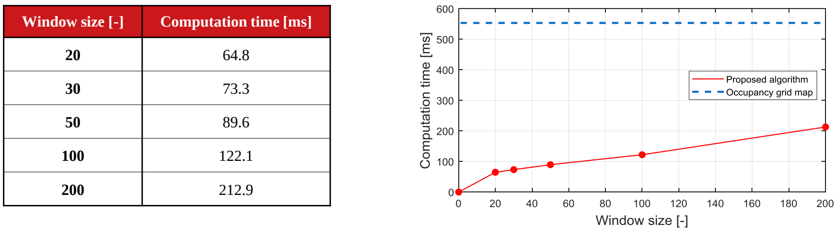
Table 2. Confusion matrices of the tracking-based algorithm and the proposed algorithm with various time window configurations.

			Actual class		**Performance**	
			Static	**Dynamic**		
Predicted class	**Tracking**	**Static**	670066	15335	Accuracy:	73%
		Dynamic	266600	93127	Static Recall:	72%
		Unknown	0	0	Dynamic Recall:	86%
	Proposed algorithm (*W* = *20*)	**Static**	490395	11589	Accuracy:	78%
		Dynamic	47926	97321	Static Recall:	82%
		Unknown	57905	51868	Dynamic Recall:	61%
	Proposed algorithm (*W* = *30*)	**Static**	506600	3257	Accuracy:	83%
		Dynamic	53631	132373	Static Recall:	83%
		Unknown	48605	25148	Dynamic Recall:	82%
	Proposed algorithm (*W* = *50*)	**Static**	490736	2863	Accuracy:	86%
		Dynamic	57348	153552	Static Recall:	83%
		Unknown	43313	4421	Dynamic Recall:	95%
	Proposed algorithm (*W* = *100*)	**Static**	480355	10662	Accuracy:	84%
		Dynamic	60341	146936	Static Recall:	82%
		Unknown	44241	2050	Dynamic Recall:	92%

To quantitatively analyze the segmentation performance, confusion matrices for the tracking-based classification algorithm and the proposed algorithm with various time window configurations were created, as shown in Table 2. Using the occupancy grid map, static point segmentation is possible, but dynamic point classification is not possible. Therefore, the performance of the segmentation algorithm based on the occupancy grid map is not included in the confusion matrix. For the segmentation using the proposed algorithm, the point motion is classified as static or dynamic when the belief of static and dynamic is over 0.8, respectively. The true class of points used as a reference for evaluation was classified manually. Although public data is more appropriate for comparing performance with other algorithms, there is no public data labeled by point-wise motion to verify the performance of real-time motion classification. The object tracking-based point motion segmentation algorithm segments a point as *dynamic* when it is located inside the bounding box of the track above a certain speed, and the remaining points not included in the moving track are segmented as *static*. However, the performance of point-wise motion segmentation is not superior due to incorrect bounding boxes and inaccurate speed estimation by the tracker. The segmentation accuracy of the proposed algorithm is better than that of the tracking-based segmentation algorithm when the time window W is 20, 30, 50, and 100, as shown in Table 2. The segmentation performance for a time window of 50 is better than that of 100 because the drift error of pose estimation affects the segmentation performance.

6.3. Real-Time Performance Evaluation

The algorithm was verified in an RTMaps environment with a QuadCore Intel Core i5-3570K, 3600 MHz (36 × 100) CPU. Hard real-time performance could not be fully evaluated because it is not an embedded environment, but it can be optimized later by checking the soft real-time performance in the RTMaps environment. To evaluate the real-time performance of the algorithm, the occupancy grid map-based segmentation algorithm was compared with the proposed algorithm. As shown in Figure 12, the algorithm based on occupancy grid maps took a long time because all cells in the area the LiDAR beam passed were constantly updated. The larger the window size of the proposed algorithm, the more computation is required. The most appropriate time window setting for the proposed algorithm is 50, as illustrated by the confusion matrix, and its computation time is below the sampling period of the Velodyne LiDAR (100 milliseconds).



Window size [-]	Computation time [ms]
20	64.8
30	73.3
50	89.6
100	122.1
200	212.9

(a) (b)

Figure 12. (**a**) table and (**b**) plot of Computation time for the occupancy grid map and the proposed algorithm.

7. Conclusions

This paper proposed a segmentation algorithm to rapidly classify the motion states of a LiDAR point cloud in real-time. The motion segmentation algorithm requires inputs of point clouds and 3D pose (position and direction) of the LiDAR sensor. The point-wise motion segmentation is performed based on the laser beam characteristics and the 3D pose correlation between consecutive LiDAR points. A combination of probability and evidence theory is used to accurately and reliably segment the motion state of points into *dynamic*, *static*, and *unknown*.

(1) The point motion segmentation algorithm considers the characteristics of the LiDAR laser beam, such as multi-echo, beam divergence, and horizontal and vertical resolution. Therefore, the point motions are segmented more accurately and reliably than by conventional algorithms (e.g., occupancy grid mapping and tracking-based segmentation algorithm).

(2) To update the motion state of each LiDAR point, a combination of probability theory and evidence theory is applied to point motion modeling to accurately reflect the LiDAR characteristics. Probability theory is used to model the likelihood fields of LiDAR point clouds, taking into account the uncertainty of LiDAR measurements. Evidence theory is used to incorporate multipoint motion probabilities, taking into account pose uncertainty and the *unknown* state.

(3) The proposed point motion segmentation algorithm was evaluated experimentally. The segmentation accuracy was 86% for a time window of $W = 50$. This is better than the accuracy of the tracking-based algorithm (73%). Because the proposed algorithm can handle one LiDAR point

clouds in a one-step process, when operating under 100 ms, it is suitable for real-time applications in automated and intelligent vehicle systems.

The performance of the algorithm is related to the positioning algorithm that estimates the pose of the LiDAR sensor. In future research, we plan to analyze the quantitative relationship between the LiDAR sensor positioning and the proposed algorithm performance, and to study the SLAM algorithm that classifies the point motion and simultaneously estimates the pose of the LiDAR sensor.

Author Contributions: Conceptualization, K.J.; Methodology, K.J.; Software, K.J. and S.L. and C.K.; Validation, K.J. and S.L.; Formal Analysis, S.L.; Investigation, K.J.; Resources, K.J.; Data Curation, S.L.; Writing—Original Draft Preparation, K.J.; Writing—Review & Editing, K.J. and M.S.; Visualization, S.L.; Supervision, K.J. and M.S.; Project Administration, K.J. and M.S.; Funding Acquisition, K.J.

Funding: This paper was supported by Konkuk University in 2018.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Goodin, C.; Carruth, D.; Doude, M.; Hudson, C.; Goodin, C.; Carruth, D.; Doude, M.; Hudson, C. Predicting the Influence of Rain on LIDAR in ADAS. *Electronics* **2019**, *8*, 89. [CrossRef]
2. Takai, I.; Matsubara, H.; Soga, M.; Ohta, M.; Ogawa, M.; Yamashita, T.; Takai, I.; Matsubara, H.; Soga, M.; Ohta, M.; et al. Single-Photon Avalanche Diode with Enhanced NIR-Sensitivity for Automotive LIDAR Systems. *Sensors* **2016**, *16*, 459. [CrossRef] [PubMed]
3. Cui, T.; Ji, S.; Shan, J.; Gong, J.; Liu, K.; Cui, T.; Ji, S.; Shan, J.; Gong, J.; Liu, K. Line-Based Registration of Panoramic Images and LiDAR Point Clouds for Mobile Mapping. *Sensors* **2016**, *17*, 70. [CrossRef] [PubMed]
4. Im, J.H.; Im, S.H.; Jee, G.I.; Im, J.H.; Im, S.H.; Jee, G.I. Extended Line Map-Based Precise Vehicle Localization Using 3D LIDAR. *Sensors* **2018**, *18*, 3179. [CrossRef] [PubMed]
5. An, J.; Choi, B.; Sim, K.B.; Kim, E.; An, J.; Choi, B.; Sim, K.B.; Kim, E. Novel Intersection Type Recognition for Autonomous Vehicles Using a Multi-Layer Laser Scanner. *Sensors* **2016**, *16*, 1123. [CrossRef] [PubMed]
6. Thrun, S.; Burgard, W.; Fox, D. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*; The MIT Press: Cambridge, MA, USA, 2005.
7. Pagac, D.; Nebot, E.M.; Durrant-Whyte, H. An evidential approach to probabilistic map-building. In Proceedings of the IEEE International Conference on Robotics and Automation, Minneapolis, MN, USA, 22–28 April 1996; Volume 1, pp. 745–750. [CrossRef]
8. Alsayed, Z.; Bresson, G.; Nashashibi, F.; Verroust-Blondet, A. *PML-SLAM: A Solution for Localization in Large-Scale Urban Environments*; HAL: Bengaluru, India, 2015.
9. Jo, K.; Cho, S.; Kim, C.; Resende, P.; Bradai, B.; Nashashibi, F.; Sunwoo, M. Cloud Update of Tiled Evidential Occupancy Grid Maps for the Multi-Vehicle Mapping. *Sensors* **2018**, *18*, 4119. [CrossRef] [PubMed]
10. Trehard, G.; Pollard, E.; Bradai, B.; Nashashibi, F. On line mapping and global positioning for autonomous driving in urban environment based on evidential SLAM. In Proceedings of the 2015 IEEE Intelligent Vehicles Symposium (IV), Seoul, Korea, 28 June–1 July 2015. [CrossRef]
11. Moras, J.; Cherfaoui, V.; Bonnifait, P. Moving objects detection by conflict analysis in evidential grids. In Proceedings of the IEEE Intelligent Vehicles Symposium, Baden-Baden, Germany, 5–9 June 2011; pp. 1122–1127. [CrossRef]
12. Moras, J.; Cherfaoui, V.; Bonnifait, P. A lidar perception scheme for intelligent vehicle navigation. In Proceedings of the 11th International Conference on Control Automation Robotics Vision, Singapore, 7–10 December 2010; pp. 1809–1814. [CrossRef]
13. Moras, J.; Cherfaoui, V.; Bonnifait, P. Credibilist occupancy grids for vehicle perception in dynamic environments. In Proceedings of the IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 84–89. [CrossRef]
14. Huang, R.; Chen, J.; Liu, J.; Liu, L.; Yu, B.; Wu, Y.; Huang, R.; Chen, J.; Liu, J.; Liu, L.; et al. A Practical Point Cloud Based Road Curb Detection Method for Autonomous Vehicle. *Information* **2017**, *8*, 93. [CrossRef]
15. Dimitrievski, M.; Veelaert, P.; Philips, W.; Dimitrievski, M.; Veelaert, P.; Philips, W. Behavioral Pedestrian Tracking Using a Camera and LiDAR Sensors on a Moving Vehicle. *Sensors* **2019**, *19*, 391. [CrossRef] [PubMed]

16. Kim, C.; Cho, S.; Sunwoo, M.; Jo, K. Crowd-Sourced Mapping of New Feature Layer for High-Definition Map. *Sensors* **2018**, *18*, 4172. [CrossRef] [PubMed]
17. Kim, D.; Jo, K.; Lee, M.; Sunwoo, M. L-Shape Model Switching-Based Precise Motion Tracking of Moving Vehicles Using Laser Scanners. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 598–612. [CrossRef]
18. Petrovskaya, A.; Thrun, S. Model based vehicle detection and tracking for autonomous urban driving. *Auton. Robot.* **2009**, *26*, 123–139. [CrossRef]