

Article

Motion Field Estimation for a Dynamic Scene Using a 3D LiDAR

Qingquan Li ^{1,2}, Liang Zhang ^{1,5,*}, Qingzhou Mao ^{1,*}, Qin Zou ^{3,*}, Pin Zhang ⁴, Shaojun Feng ⁵
and Washington Ochieng ⁵

¹ State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, No.129 Luoyu Road, Wuhan 430079, China; E-Mail: qqli@whu.edu.cn

² Shenzhen Key Laboratory of Spatial-temporal Smart Sensing and Services, Shenzhen University, No.3688 Nanhai Road, Shenzhen 518060, China

³ School of Computer, Wuhan University, No.129 Luoyu Road, Wuhan 430072, China

⁴ School of Geodesy and Geomatics, Wuhan University, No.129 Luoyu Road, Wuhan 430079, China; E-Mail: zhangpin.whu@gmail.com

⁵ Centre for Transport Studies (CTS), Imperial College London, Exhibition Road, London SW7 2AZ, UK; E-Mails: s.feng@imperial.ac.uk (S.F.); w.ochieng@imperial.ac.uk (W.O.)

* Authors to whom correspondence should be addressed; E-Mails: zhang.liangtrc@gmail.com (L.Z.); qzhmao@whu.edu.cn (Q.M.); qinnzou@gmail.com (Q.Z.); Tel.: +86-027-6877-0625 (L.Z.); +86-027-6877-9677 (Q.M.); +86-027-6877-5721 (Q.Z.).

Received: 16 July 2014; in revised form: 26 August 2014 / Accepted: 3 September 2014 /

Published: 9 September 2014

Abstract: This paper proposes a novel motion field estimation method based on a 3D light detection and ranging (LiDAR) sensor for motion sensing for intelligent driverless vehicles and active collision avoidance systems. Unlike multiple target tracking methods, which estimate the motion state of detected targets, such as cars and pedestrians, motion field estimation regards the whole scene as a motion field in which each little element has its own motion state. Compared to multiple target tracking, segmentation errors and data association errors have much less significance in motion field estimation, making it more accurate and robust. This paper presents an intact 3D LiDAR-based motion field estimation method, including pre-processing, a theoretical framework for the motion field estimation problem and practical solutions. The 3D LiDAR measurements are first projected to small-scale polar grids, and then, after data association and Kalman filtering, the motion state of every moving grid is estimated. To reduce computing time, a fast data association algorithm is proposed. Furthermore, considering the spatial correlation of motion among neighboring grids, a novel spatial-smoothing algorithm is also presented to optimize the motion field. The experimental

results using several data sets captured in different cities indicate that the proposed motion field estimation is able to run in real-time and performs robustly and effectively.

Keywords: 3D LiDAR; motion field estimation; motion sensing; spatial smoothing

1. Introduction

The accurate perception of the motion of moving objects is a key technology for active collision avoidance systems and intelligent driverless vehicle systems. Multi-target tracking (MTT) ([1–5]) is one effective method to accomplish this work, which estimates the motion state of all of the detected targets, such as vehicles, pedestrians, and so on. Another emerging technology is visual motion field estimation (VMFE) ([6–15]), which estimates the motion state of each point using a camera system.

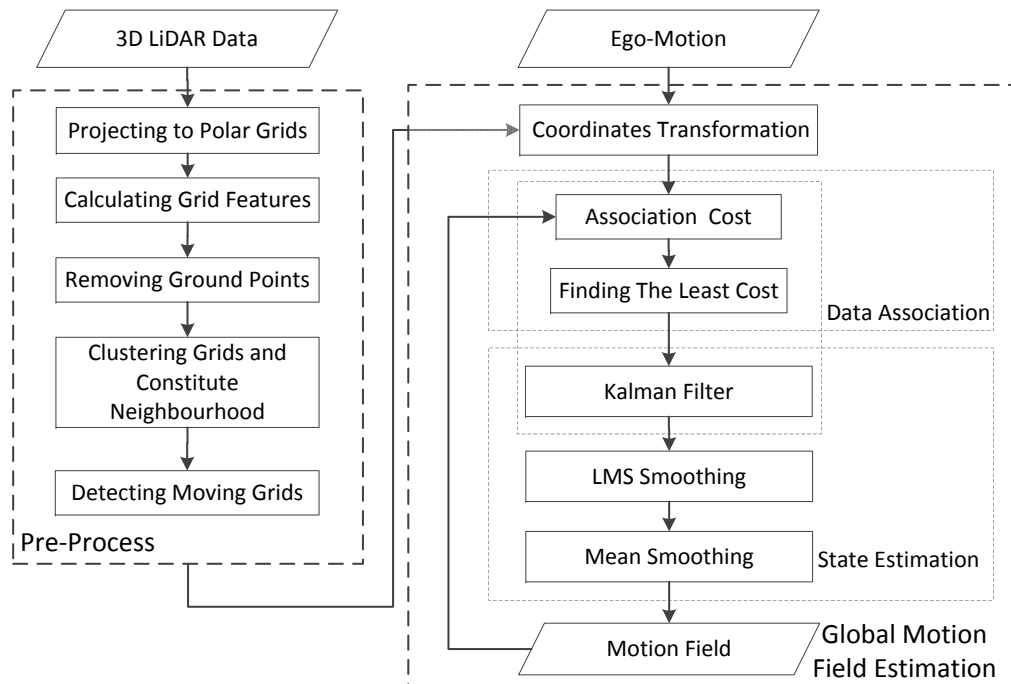
MTT is a well-theorized and intuitive motion sensing method. It includes three main steps, that is object segmentation, data association and motion estimation. In the object segmentation step, meaningful measurements, such as vehicles and pedestrians, are segmented out from raw data, and in the data association step, the relationship between these measurements and tracked targets is determined. Following this, in the motion estimation step, the motion state of each target is estimated by Kalman filter or extended Kalman filter. In recent years, 3D LiDAR has been widely used by intelligent driverless vehicles for environmental sensing, and researchers proposed a variety of 3D LiDAR-based MTT methods. Petrovskaya and Thrun [1] presented a model-based vehicle tracking algorithm using a 3D LiDAR. In their algorithm, they represented a vehicle by a 2D model, that is, a rectangle with a width and a length, and assumed the shape of the model to be invariant with time. They tried to use the temporal constraint of the shape to overcome the occlusion problem, which would affect the accuracy and consistency of tracking. However, this would lead to another problem, that is the initialization error of the model. Shackleton *et al.* [2], Kalyan *et al.* [3] each proposed similar pedestrian tracking methods based on the 3D LiDAR. In their method, the pedestrian was first segmented out and then modeled as a 3D model, that is a cylinder or cube. Liang *et al.* [4] proposed a practical framework for vehicle-like target tracking, which integrated dynamic point cloud registration (DPCR) and multiple hypothesis tracking (MHT, [16–18]). In that framework, the moving parts of raw measurements were simply segmented into vehicle-like targets first and represented by rectangles. The DPCR was developed to calculate the real-time ego-pose accurately; meanwhile, MHT tracked the vehicle-like targets consistently and helped to improve the performance of DPCR by discriminating and removing the dynamic measurements. In the above methods, moving objects have to be detected and modeled before tracking. Although the temporal constraint on model shape can be used to cope with partial occlusion to some degree, it also leads to some new problems, such as inaccurate motion estimation and limitations in some special target models. Moosmann and Stiller [5] proposed a joint self-localization and tracking of arbitrary objects method using the 3D LiDAR, in which they combined dynamic data partitioning with track before detect (TBD, [19–26]) techniques. Their method does not need to model the objects and is therefore able to track arbitrary objects. Furthermore, with the TBD technique, it is possible to detect moving objects

robustly. At about 5 m/s, however, the accuracy of the motion state calculated by their method is not high. Since MTT, whether based on a 3D LiDAR sensor or a camera, needs to do object segmentation first, it is subject to unavoidable segmentation errors, which result in inaccurate motion estimation. Meanwhile, if a target is associated with an incorrect measurement in the data association step, an abnormal motion state will be returned.

Unlike MTT, the MFE method determines the motion state of each point in the scene. Visual motion field estimation (VMFE) has been researched for many years in the computer vision field Driessen and Biemond [6], Dufaux and Moscheni [7], Hosur and Ma [8], Farnebäck [9], Franke *et al.* [10], Rabe *et al.* [11], Rabe *et al.* [12], Sellent *et al.* [13], Xu *et al.* [14], Hadfield [15]. MFE is able to perceive the dynamic scene fully and robustly and to track all of the moving objects without prior error-prone segmentation and detection. Furthermore, VMFE is able to correct the motion state error caused by data association errors using spatial correlation. Although most visual MFE algorithms are 2D, without depth information, in recent years, some 3D VMFE algorithms have been proposed. Franke *et al.* [10], Rabe *et al.* [11] and Rabe *et al.* [12] proposed the 6D-vision algorithm, which adopts a stereo camera system and estimates the 3D position and 3D velocity of each image point. Hadfield [15] also proposes a 3D visual MFE for natural human action recognition. The main disadvantage of 3D VMFE is that the visual sensors cannot output 3D position directly. When recovering 3D position from 2D images, position noise will be generated, which commonly leads to a noisy motion field estimation [12]. Additionally, the visual system is susceptible to the influence of environmental conditions, such as weather, illumination, shadow, and so on.

Compared to the visual system, the 3D LiDAR has two main advantages: firstly, it is not susceptible to environmental conditions; secondly, it directly outputs the 3D position of points, minimizing the position noise. However, there are few articles researching MFE using a 3D LiDAR, so far. In this paper, a novel 3D LiDAR-based MFE is proposed. Unlike well-organized visual measurements, the 3D LiDAR measurements are millions of non-organized scatter points without color information. To conduct MFE with the 3D LiDAR, the raw measurements need to be re-organized, and a new data association method different from the ones adopted in VMFE is required. In our method, the 3D LiDAR measurements are organized as small-scale polar grids, and the motion state of each grid that is potentially moving is estimated by the Kalman filter. In this way, a motion field is constituted using the 3D LiDAR measurements. To enable the method to run in real time, a fast data association algorithm is proposed. Furthermore, to optimize the motion field, a spatial-smoothing algorithm based on the spatial correlation of motion is also proposed. With the spatial-smoothing algorithm, some abnormal motion states caused by data-association errors are corrected, and the overall accuracy of the motion field is improved. The overall flow of the proposed 3D LiDAR-based MFE is given in Figure 1.

The remainder of this paper is organized as follows: Section 2 describes the pre-processing of raw measurements; Section 3 details the global motion field estimation algorithm, including data association, motion state estimation and spatial smoothing; Section 4 demonstrates the experimental results in real-world scenarios; Section 5 offers some conclusions and remarks.

Figure 1. The overall algorithm flow.

2. Pre-Processing for 3D LiDAR Measurements

The pre-processing includes projection from raw measurements to horizontal grids, removal of ground points and constitution of neighborhood systems.

The 3D LiDAR adopted in this paper is the Velodyne HDL-64D LiDAR [27], which scans the environment with a 360 degree horizontal field of view, a 26.8 degree vertical field of view up to 15 Hz and captures more than 1.3 million points per second (Figure 2). Since the raw measurements consist of millions of scatter points, it is impossible to do MFE directly. We therefore re-organize these scatter points by projecting them to regular grids in a horizontal direction. The grid is the basic element of the motion field. This process is reasonable for the vertical velocity of targets in urban environments, which is close to zero. After the projection, grids belonging to the ground are removed, since these are assumed to be static. Another basic idea of MFE is that neighboring grids have a similar motion state, whether they are moving or static. Thus, all of the non-ground grids are clustered to constitute neighborhood systems. The neighborhood system is the basic unit for spatial smoothing.

2.1. Projection to Grids

Considering that the density of measurements decreases as range increases, a polar (Figure 3) rather than rectangular grid is adopted in this paper. The resolution of the polar grid is adjustable, and in this paper, the angular resolution is set as one degree and the distance resolution as 0.25 m.

Each grid is denoted by g_i , where i is the serial number in the grid-field. The average height \bar{H} , the height variance σ_H^2 and the difference, ΔH , between the maximum and minimum heights are calculated as grid features. Meanwhile, the barycentric coordinates $p = (g_x, g_y)$ of each grid are also computed.

Figure 2. The 3D LiDAR data. Trees are bounded by white cubes, vehicles by green cubes and pedestrians by red cubes. The black zones are the blind spots of LiDAR.

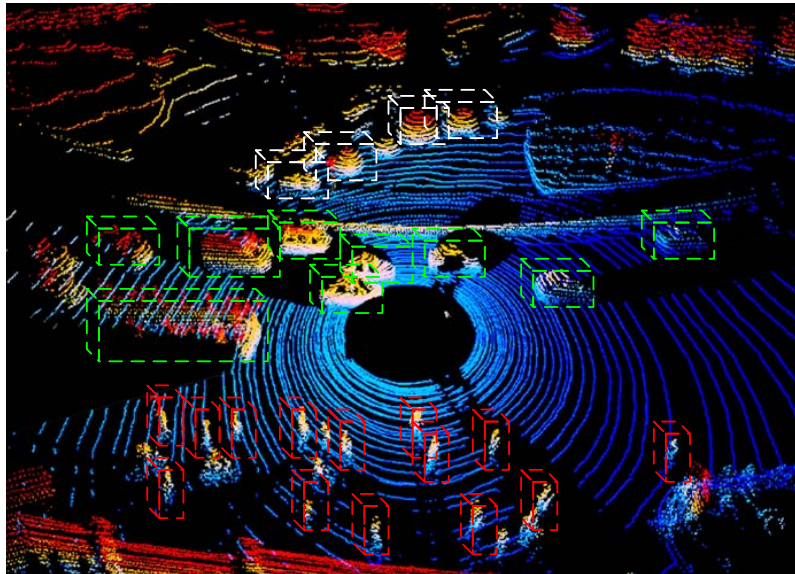
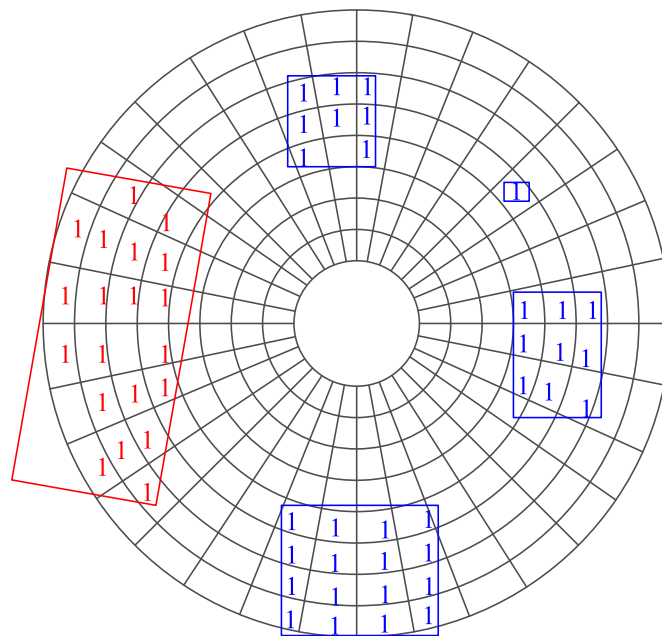


Figure 3. Constitution of neighborhood systems and labeling moving grids. The non-ground grids are clustered first to constitute the neighborhood system. If the size of a neighborhood system is larger than a pre-defined size, then it is labeled as a static system (marked by red in the figure). Otherwise, it is marked as a moving system (marked by blue in the figure).



2.2. Elimination of Ground Grids

The grid feature vector $f = [\bar{H}, \sigma_H^2, \Delta H]$ is used for classifying the grids as ground grids or non-ground grids by using Algorithm 1.

Algorithm 1 Segmentation: ground.

```

if  $|\bar{H}_i - H_g| > \delta_1 \parallel (\sigma_H^2)_i > \delta_2 \parallel \Delta H_i > \delta_3$  then
    Grid  $i$  belongs to the non-ground object;
else
    Grid  $i$  belongs to the ground.
end if

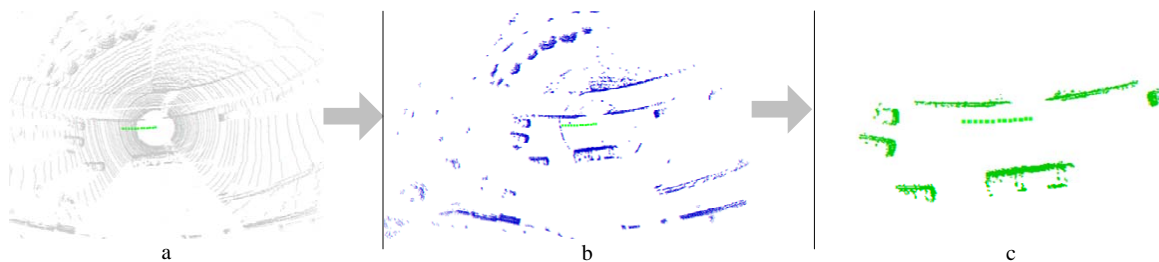
```

where H_g is the height of ground in the vehicle coordinate system, which has to be calibrated in advance, and δ_1 , δ_2 and δ_3 are thresholds, the values of which are set based on experimental statistics.

2.3. Constitution of Neighborhood Systems

The non-ground grids are clustered based on the eight-neighborhood criteria. The grids located in the same neighborhood system (denoted by Θ) are supposed to have a similar (maybe not the same) motion state. If the size of a neighborhood system is too big (in this paper, the length threshold is 12 m, and the width threshold is 9 m), then all of the grids belonging to it are marked as “static”; otherwise, the grids are marked as “moving” (Figure 3). Only the motion state of moving grids is estimated in the following process. Note that the neighborhood system constituted here is for detecting those possible moving grids and for spatial smoothing, but not for tracking (as MTT does). Figure 4 shows the flow of the pre-processing.

Figure 4. Detection of potential moving measurements. The green line is the ego-trajectory. (a) The raw measurements; (b) the non-ground measurements; (c) the potential moving measurements.



3. Motion Field Estimation

The core problem with MFE is how to estimate and optimize the motion state of each grid. In this section, the theoretical framework for MFE is deduced first based on a Bayesian formulation. The framework contains three joint problems: data association, state filtering and spatial smoothing. Solutions for these three problems are presented below.

3.1. Bayesian Framework for Global MFE

In this paper, the discrete time index is denoted by the variable k , the observation consisting of all of the moving grids at time k by $z_k = [p, f]$, all observations up to and including time k by $Z_k = [z_1 \dots z_k]$,

the motion field that contains n motion state vectors by $M_k = \{m^1, m^2, \dots, m^n\}$ and the data association by J_k . Each m^i consists of 2D planar coordinates $[\hat{g}_x, \hat{g}_y]$ and corresponding velocities $[v_x, v_y]$. If the 3D LiDAR is moving, either a positioning and orientation system (POS) or dynamic point cloud registration (DPCR) can be used to estimate the ego-pose. Equation (1) gives the Bayesian framework for MFE.

$$\begin{aligned} \underset{M_k, J_k}{Max} \quad & P(M_k, J_k | Z_k) \\ P(M_k, J_k | Z_k) = & P(J_k | Z_k) P(M_k | Z_k, J_k) \end{aligned} \quad (1)$$

As shown in Equation (1), MFE consists of two main parts, that is, the data association problem $P(J_k | Z_k)$ and the motion estimation problem $P(M_k | Z_k, J_k)$.

3.2. Data Association

Data association is generally regarded as the most complex problem in both MTT and MFE. It identifies which observations at different times belong to the same target. In general, data association is considered to be a 2D linear integer assignment problem (LIAP) and is mathematically represented as Equation (2):

$$\begin{aligned} \min \quad & \sum_{i=1}^N \sum_{j=1}^N \delta_{ij} C_{ij} \\ \text{s.t.} \quad & \sum_{i=1}^N \delta_{ij} = 1 \quad \text{and} \quad \sum_{j=1}^N \delta_{ij} = 1 \end{aligned} \quad (2)$$

$$\quad \quad \quad (3)$$

Here, N is the larger value of the numbers of moving grids in the previous and current scans. C_{ij} is the cost when associating the grid j of the last scan (denoted by g_j^L) with the grid i of the current one (denoted by g_j^C). $\delta_{ij} \in \{0, 1\}$, if g_j^L is assigned to g_j^C , $\delta_{ij} = 1$, otherwise, $\delta_{ij} = 0$. $P(J_k | Z_k)$ is an exponential function regarding $\sum C_{ij}$. In VMFE, the cost C_{ij} is calculated based on visual features, such as scale-invariant feature transform (SIFT, [28,29]). Since the projected grid has only three features, it would be difficult to perform the association just using grid features. To solve this problem, the TBD technique is applied here: the position of measurements is predicted using the previous motion state, and as a result, the searching range of association will be minimized. According to the Bayesian formulation,

$$P(J_k | Z_k) \propto P(J_k) P(Z_k | J_k) \propto - \sum C_{ij} \quad (4)$$

$P(Z_k | J_k)$ is the likelihood function relative to observations $[p, f]$. Unlike multiple hypothesis tracking, which adopts suboptimal data association solutions, as well as the optimal one, our algorithm only adopts the optimal data association solution at each iteration, so the previous data association has no significance on current data association and the priority $P(J_k)$ is a constant at each iteration. Besides, the observations of the last scans have no direct influence on current data association either. Thus, the cost C_{ij} just depends on current observations and the predicted motion state. More specifically, the cost is constituted as two parts, with the first being relative to position difference between measured position and predicted position and the other relative to the feature difference. Mathematically,

$$C_{ij} = \alpha \Phi(p_i, p_j) + \beta \Psi(f_i, f_j) \quad (5)$$

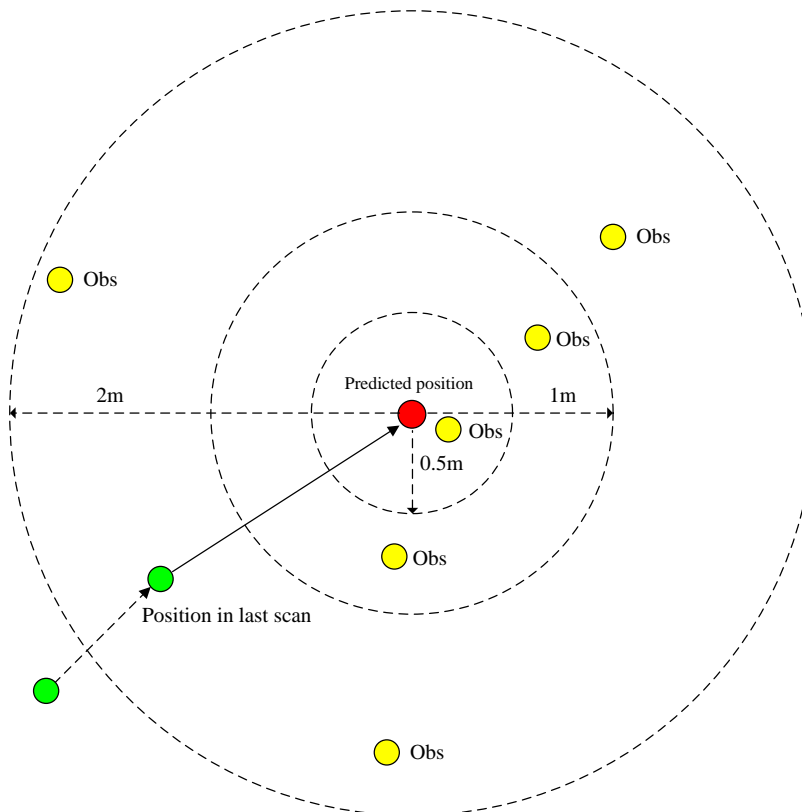
where i is the index of a moving grid g_i in the current scan and j is the index of a moving grid g_j in the last scan. p_i is the measured position of g_i , and p_j is the predicted position of g_j in the current scan. f_i and f_j are the feature vectors of g_i and g_j , respectively. $\Phi(p_i, p_j)$ and $\Psi(f_i, f_j)$ are cost functions regarding position difference and feature difference, respectively, and α and β are the weight values ranging from zero to one. Specifically, the cost is computed using Equation (6),

$$\begin{aligned}\Phi(p_i, p_j) &= \Delta l_{ij} \\ \Psi(f_i, f_j) &= \overline{H}_{ij} + \sigma_{H_{ij}} \\ C_{ij} &= \alpha \Delta l_{ij} + \beta (\overline{H}_{ij} + \sigma_{H_{ij}}^2)\end{aligned}\quad (6)$$

where Δl_{ij} is the Euclidean distance between p_i and p_j , \overline{H}_{ij} is the difference in average height and $\sigma_{H_{ij}}$ is the difference in the standard deviation of height. Since the value of difference between maximum and minimum heights is unstable, it is not used here. If Δl_{ij} is bigger than a threshold δ_l (Equation (7), Figure 5), the C_{ij} is set as $+\infty$ (in practice, C_{ij} is set as a very big value).

$$\delta_l = \begin{cases} 2\text{ m} & \text{if the grid } j \text{ has not been tracked} \\ 1\text{ m} & \text{if the grid } j \text{ has been tracked once} \\ 0.5\text{ m} & \text{if the grid } j \text{ has been tracked twice or more} \end{cases} \quad (7)$$

Figure 5. Association threshold.



The computational complexity of the constitution of the cost matrix is $O(N^2)$. After the cost matrix is constituted, the rest of the problem is to solve the optimum assignment matrix from it. There are many algorithms to solve the LIAP, such as the Auction algorithm [30] and the JVLAP algorithm [31].

However, all of the computational complexity of these algorithms are $O(N^3)$; so, the total computational complexity of the data association will be $O(N^3)$, and it will therefore be very time consuming to solve the assignment matrix when the number of moving grids is large. To reduce the computation time, this paper proposes a relaxation method for the LIAP, in which the assignment matrix is solved at the same time that the cost matrix is generated, so that the computational complexity becomes $O(N^2)$. Accordingly, we relaxed the LIAP as in Equation (8):

$$\begin{aligned} & \min \sum_{j=1}^N \delta_{ij} C_{ij} \\ \text{s.t. } & \sum_{j=1}^N \delta_{ij} = 1, \forall i \in [1, M] \end{aligned} \quad (8)$$

where N is the number of moving grids in the last scan and M is the number of moving grids in the current scan. Equation (8) indicates that each moving grid g_i in the current scan can be associated with one moving grid at most in the last scan with the local least association cost, but that a moving grid g_j in the last scan may be assigned to several moving grids in the current scan. The relaxation of the LIAP is locally optimal, but not globally optimal, so it may result in some association errors. In practice, however, most of the associated grid-pairs are correct, so the association errors can be corrected by the smoothing process in Section 3.3.2. Algorithm 2 gives the flow of relaxation of the LIAP. Note that, in this algorithm, the motion state of each moving grid in the current scan is updated immediately when it is associated with a moving grid in the last scan, which also reduces the whole time cost.

Algorithm 2 Relaxation of the linear integer assignment problem (LIAP).

Require:

M moving grids of current scan

N moving grids of last scan

for all Moving Grid i of current scan **do**

$Mincost = +\infty$

for all Moving Grid j of last scan **do**

Predicting the position of Grid j in current scan

if Predicted position of Grid j falling in association gate of Grid i **then**

Calculating association cost C_{ij}

if $C_{ij} < Mincost$ **then**

$Mincost = C_{ij}$

$\delta_{ij}=1$

end if

end if

end for

Grid i 's motion state is updated using the Kalman filter in Section 3.3.1.

end for

3.3. Motion State Estimation and Spatial Smoothing

Following data association, the remaining problems are to estimate the motion state of each grid and to optimize the motion field. According to the Bayesian framework 1, the problem of global motion state estimation is to solve Equation (9):

$$\underset{M_k}{Max} P(M_k|J_k, Z_k) \quad (9)$$

The data association J_k has been determined in the data association step. In general, the motion state is regarded as a one-order Gauss–Markov process. As a result, the motion state estimation becomes Equation (10),

$$P(M_k|J_k, Z_k) = \frac{1}{c} P(M_k|M_{k-1}) P(J_k, Z_k|M_k) \quad (10)$$

where $\frac{1}{c}$ is a proportionality constant. In the motion field, a grid's motion state is assumed to be affected by its neighbors, that is the neighboring grids have similar motion properties. Adding this spatial correlation into Equation (10), the problem of global motion estimation becomes Equation (11):

$$\underset{M_k}{Max} P(M_k|M_{k-1}) P(J_k, Z_k|M_k) P(M_k|\Theta_k) \quad (11)$$

where Θ_k is the set of neighborhoods. As formulated in Equation (11), the motion estimation problem includes three steps: motion prediction $P(M_k|M_{k-1})$, motion update with measurements $P(J_k, Z_k|M_k)$ and motion smoothing $P(M_k|\Theta_k)$. The prediction process and update process together constitute the filtering process.

3.3.1. Filtering Process

The position observations p consist of measured barycentric coordinates $([g_x, g_y]^T)$, and the filtered state m consists of the filtered position and velocity, which is denoted by $[\hat{g}_x, \hat{g}_y, v_x, v_y]^T$. In the filtering step, the linear Kalman filter with a constant velocity model is applied. A Kalman filter is generated for each moving grid in the current scan and then initialized with the measured coordinates and zero velocity. If a moving grid g_i in the current scan is successfully associated with a moving grid g_j in the last scan, its Kalman filter F_i is replaced by the last Kalman filter K_j and then updated with the observation of g_i 's barycentric coordinates. The prediction process and update process are respectively modeled as Equation (12):

$$\begin{aligned} m_k &= A m_{k-1} + v_{k-1} \\ p_k &= H m_k + w_k \end{aligned} \quad (12)$$

where A and H are the state transition matrix and the observation matrix, respectively, both of which are time invariant; v_k and w_k are both white Gaussian noise sequences with zero means. Specifically,

$$A = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (13)$$

where Δt is the scanning period of the 3D LiDAR.

3.3.2. Spatial Smoothing Process

Because of data association and observation errors, the motion field calculated through the filtering process may not be spatially coherent. Spatial smoothing is based on the spatial coherence hypothesis: that the neighboring grids have similar velocities. Please note that, in MFE, we just need to correct the velocity, since the position of a moving grid is accurate enough.

An improved least mean square-based spatial smoother (LMS-smoothing) ([32,33]) is first applied to optimize the motion field. The velocity vector of a moving grid g_0 is denoted by v_0 , the velocity vector of its neighboring cell by v_i and the number of its neighboring cells by N . The mean square ξ_{g_0} of g_0 is calculated by Equation (14),

$$\xi_{g_0} = \sum_{i=1}^N \frac{1}{N} \|v_0 - v_i\| \quad (14)$$

Similarly, the mean square ξ_{g_i} of each g_i is calculated by Equation (15),

$$\xi_{g_i} = \sum_{j=0}^N \frac{1}{N} \|v_i - v_j\|, i \in [1, N], j \neq i \quad (15)$$

In [33], if Equation (16) satisfies,

$$\xi_{g_0} > \max(\xi_{g_i}) \quad (16)$$

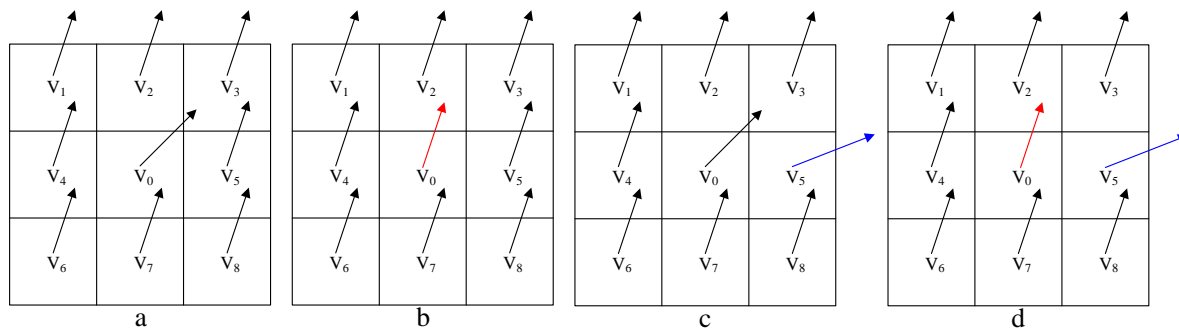
Then v_0 is replaced with the one with the least mean square (here, $\max(\xi_{g_i})$ stands for the maximum one of ξ_{g_i}). There is an underlying assumption in this method that g_0 is the only abnormal one in its neighborhood, but all of the motion states of g_0 's neighboring grids are more accurate than v_0 . This assumption is not valid all of the time, however. When there is a more abnormal grid in the neighborhood, g_0 would not be corrected. To solve this problem, we improve the criteria to Equation (17),

$$\xi_{g_0} > \min(\xi_{v_i}) \quad \text{and} \quad \frac{\min(\xi_{g_i})}{\xi_{g_0}} < 0.95 \quad (17)$$

Then v_0 is replaced with the one with the least mean square (here, $\min(\xi_{g_i})$ stands for the minimum one of ξ_{g_i}). This criteria not only ensures that all of the abnormal motion states are corrected, but it also avoids over-smoothing.

Following the LMS smoothing (Figure 6), the mean smoothing is carried out for each neighborhood system: the average velocity \bar{v} and corresponding standard deviation σ_v of a neighborhood system are computed first; if the velocity v_i of Grid i belonging to this neighborhood system does not locate in $[\bar{v} - 2\sigma_v, \bar{v} + 2\sigma_v]$, then v_i will be replaced by the average velocity \bar{v} . This mean smoothing ensures that the local anomalies within a neighborhood system will be eliminated.

Figure 6. LMS smoothing. (a) The raw filtering results. If the central grid is abnormal, but all of its neighbors are normal, it will be corrected either under the criteria of Equation (16) or the criteria of Equation (17), as shown in (b). However, if one of its neighboring grids is more abnormal (c) by v_5 , it will not be corrected under the criteria of Equation (16), but will be corrected under the proposed criteria of Equation (17) (d).



4. Experiments

In the following experiments, the performance of the proposed 3D LiDAR-based MFE is tested, including time-cost, accuracy, time-consistency and the effectiveness of the LMS-mean-smoothing.

The first data set for validation was collected with a Velodyne-64E LiDAR mounted on an intelligent driverless car (Smart V-II) developed by Wuhan University. The experimental site was located in LuXiang Circle of LuoYu Road in Wuhan, China, and the experimental time lasted 925 scans. The traffic situation in LuXiang Circle was dense and sophisticated. The top part of Figure 7 is the bird's-eye view of the circle, and the figures below show the traffic situation, which was acquired synchronously with a CCD camera. There were about 10 to 15 vehicles per scan, although the numbers were not constant. The latitudinal sensing range was set as 9 m, and the longitudinal sensing range was variable.

The other data set for validation was from the Karlsruhe Institute of Technology (KIT, [34]). In that data set, the ego-vehicle stopped at a cross-intersection, with a car in front of it, which resulted in occlusion in the point cloud of every scan (Figure 8).

The hardware setups of the computer used in our experiments were 8 GiB of memory and a 3.4-GHz i7 dual-core processor, and the software setups were Windows 8 and Visual Studio 2010.

4.1. Effectiveness of the LMS Mean Smoothing Algorithm

The performance of the proposed LMS mean smoothing algorithm is validated first. Figure 9 shows, respectively: a typical scenario when carrying out MFE with no smoothing; with the LMS smoothing presented in [33] (represented by LMS-1), with the LMS smoothing proposed in this paper (represented by LMS-2) and with the proposed LMS mean smoothing.

The standard deviation θ_i of the velocities of the moving grids belonging to a neighborhood system i is computed in order to indicate its spatial consistency. Subsequently, the average of all of the θ along the whole process (denoted by $\bar{\theta}$) is regarded as the overall index of the smoothing algorithm's performance. The experimental results are presented in Table 1.

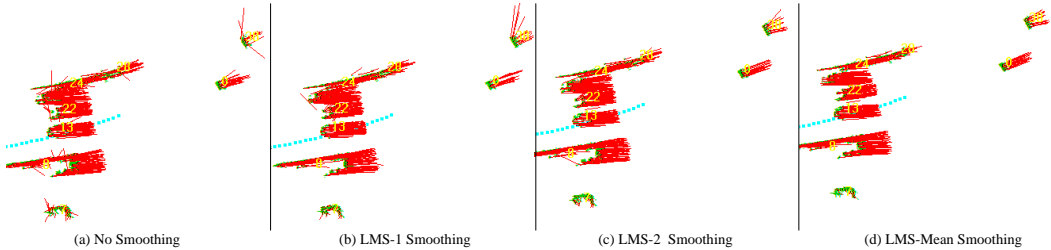
Figure 7. Bird's eye view of LuXiang Circle and typical traffic scenarios.



Figure 8. One test scenario of the Karlsruhe Institute of Technology (KIT) data set.



Figure 9. Comparison of various smoothing algorithms. (a) The raw motion field without spatial smoothing. Furthermore, comparing (b) and (c), we found that the performance of LMS-2 smoothing is better than that of LMS-1 smoothing. Obviously, the motion field smoothed by LMS mean smoothing (d) is the optimum one with the best spatial consistency.



As shown in Table 1, the $\bar{\theta}$ of the MFE with LMS mean smoothing is the smallest, at just about one quarter of that of the MFE with no smoothing. It is also clear that the proposed LMS mean smoothing outperforms LMS-2-smoothing, as well as Guo and Kim [33]’s LMS-1-smoothing and that the proposed LMS-2-smoothing performs better than the LMS-1-smoothing algorithm. After performing LMS mean smoothing, the spatial consistency of the motion field becomes much better, and almost all of the outliers are removed.

Table 1. $\bar{\theta}$ of motion field estimation (MFE) with various smoothing algorithms (unit: cm/s).

Method	Wuhan		KIT	
	x	y	x	y
No Smoothing	130.5	129.3	68.7	63.0
LMS-1	95.2	94.3	56.8	49.8
LMS-2	85.0	83.8	53.3	47.2
LMS Mean	25.3	25.2	13.5	11.5

4.2. Comparison between 3D LiDAR-Based MFE and MTT

In this subsection, we compare the performance of the proposed MFE algorithm (Figure 10b) with the MTT algorithm (Figure 10a) proposed in [4]. The pre-processing required for MTT and MFE is same. In MTT, each neighborhood system is regarded as a moving target, and its position and velocity are also filtered by the linear Kalman filter, while, in MFE, for each moving neighborhood system, the average of the velocities of its constituent grids is calculated as its mean velocity, which will be compared with the velocity of the corresponding target in MTT. Here, the accuracy and the time-consistency of the motion state estimated by MTT and MFE are compared, respectively.

The accuracy is measured by the difference with ground truth (GT) using the KIT data set, in which the ego-vehicle stopped at a cross-intersection. The length of the cross-intersection is about 18 m with a maximum error of ± 1 m, and the time of each vehicle running across the intersection is also known with a maximum error of ± 0.2 s (the average time is 2.5 s). Thus, the velocity of each car can be estimated manually with an accuracy of ± 20 cm/s, which is regarded as the ground truth. The experimental results are shown in Table 2. Please note that the velocity value presented in the tables is the one after the Kalman filter becomes stable.

Table 2. Velocities calculated by multi-target tracking (MTT) and MFE (unit: cm/s). GT, ground truth.

Method	1	2	3	4	5	6	7	8
MTT	431.5	899.8	326.9	373.5	178.6	248.8	184.7	18.4
MFE	456.6	400.4	441.3	404.6	309.2	607.5	564.2	932.5
GT	600.0	600.0	500.0	666.7	562.5	667.7	782.6	947.4

As shown in Table 2, the velocities calculated by MFE are much closer to the ground truth than the ones calculated by MTT. The overall accuracy of MFE is 151.35 cm/s, which is also better than the accuracy of the algorithm of Moosmann and Stiller [5].

The time consistency is measured by the standard deviation σ of the difference of velocities between neighboring scans along a certain time section. The velocity of the target vehicle should change smoothly, so the smaller the standard deviation is, the better the time-consistency. The Wuhan data set is used to compare the time consistency on account of its long time range, abundant targets and frequently occurring partial occlusions. The whole process is divided into three time sections, that is from zero to 300 scans, from 300 to 600 scans and from 600 to 925 scans. For each section, we chose a start scan randomly, picked out all of the vehicle targets manually from this scan and then tracked them for 50 continuous scans. The average $\bar{\sigma}$ of all of the tracked targets along these 50 scans was computed and is shown in Table 3.

Figure 10. Comparison of multi-target tracking (MTT) and MFE. (a) Scenarios for MTT in which the cyan trajectory is the ego-trajectory, the green points are the moving LiDAR points and the pink trajectories represent the trajectories of targets. (b) Scenarios for MFE in which the cyan trajectory is the ego-trajectory, the green points are the moving LiDAR points and the red lines stand for the velocity vectors of moving grids. The MTT just calculates the motion state of each detected target constituted by grid clusters, but the MFE estimates the motion state of each grid.

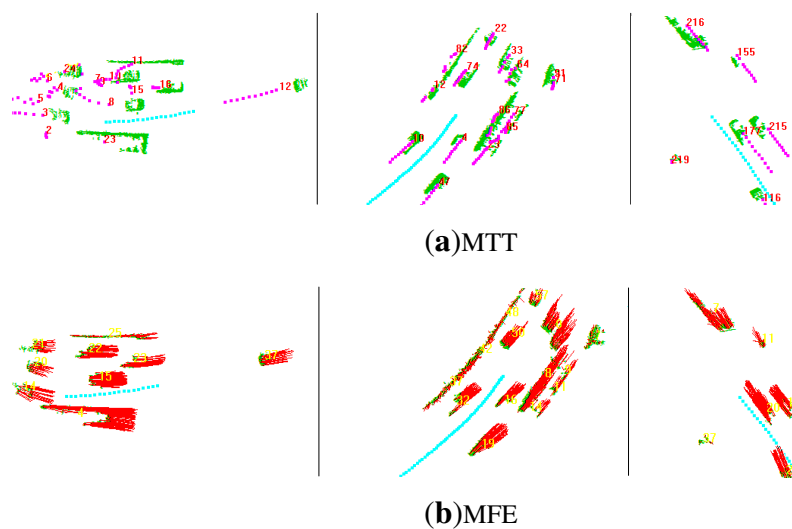


Table 3. The average standard deviation of the difference of the velocities between neighboring scans.

Method	13~62		404~453		811~860	
	x	y	x	y	x	y
MTT	216.4	80.5	31.3	63.7	38.7	67.6
MFE	22.2	16.1	38.1	50.7	11.3	8.7

According to Table 3, in general, the time consistency of the motion state estimated by MFE significantly outperforms MTT. As mentioned before, in MTT, unstable segmentation and incorrect data association may result in dramatic changes of the motion state. In contrast, the MFE does not need to perform the error-prone segmentation process and is also robust to data association errors. To give a more explicit picture, for each section, we picked out six targets (neighborhood systems)

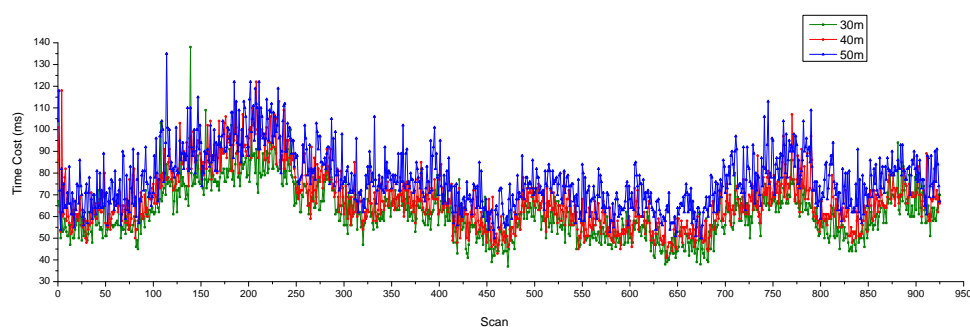
from specific continuous scans (at most, 10 continuous scans) when partial occlusions or incorrect data association happened in MTT and then calculated their velocity standard deviation along these scans. The experimental results are given in Table 4.

Table 4. Standard deviation of the difference of velocities between neighboring scans (unit: cm/s).

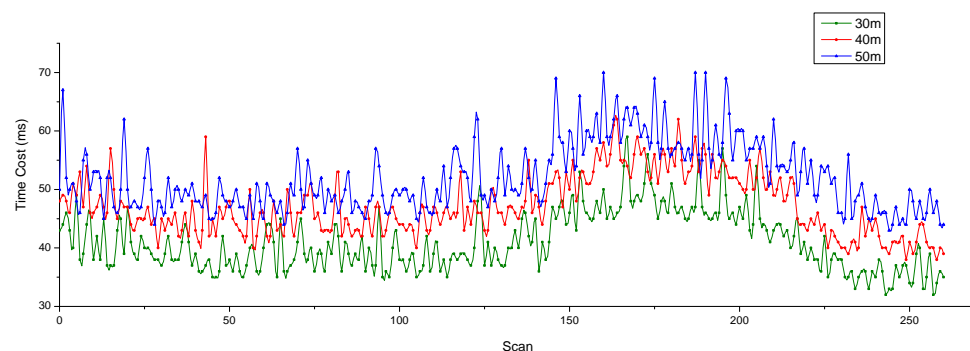
0~300					300~600					600~925				
ID	MFE		MTT		ID	MFE		MTT		ID	MFE		MTT	
	x	y	x	y		x	y	x	y		x	y	x	y
1	12.6	13.3	67.5	59.6	1	26.8	21.0	6.1	28.0	1	6.3	4.8	3.4	2.2
2	3.6	3.6	426.8	106.0	2	13.3	11.6	17.0	62.1	2	9.9	8.0	3.4	6.8
3	4.7	4.0	560.0	130.5	3	3.8	8.3	11.2	4.1	3	5.4	2.2	2.9	3.2
4	54.5	44.7	136.5	53.5	4	6.4	11.2	20.9	74.0	4	31.4	23.7	86.3	206.7
5	6.4	4.1	284.5	80.7	5	10.3	17.5	7.8	9.3	5	7.4	5.8	261.2	167.0
6	3.0	3.4	8.9	3.7	6	55.6	50.4	172.1	344.6	6	2.7	3.3	5.8	3.3

As clearly shown in Table 4, there are some large standard deviations in MTT. These large values are mainly because of the data association errors. Abnormal standard deviations tend to happen more often during the first 300 scans. This is because, at the beginning, the Kalman filters in MTT are not convergent, so data association errors are more likely to occur. As time goes on, the Kalman filters become stable, so data association errors decrease. To conclude, in general, MFE outperforms MTT on time consistency.

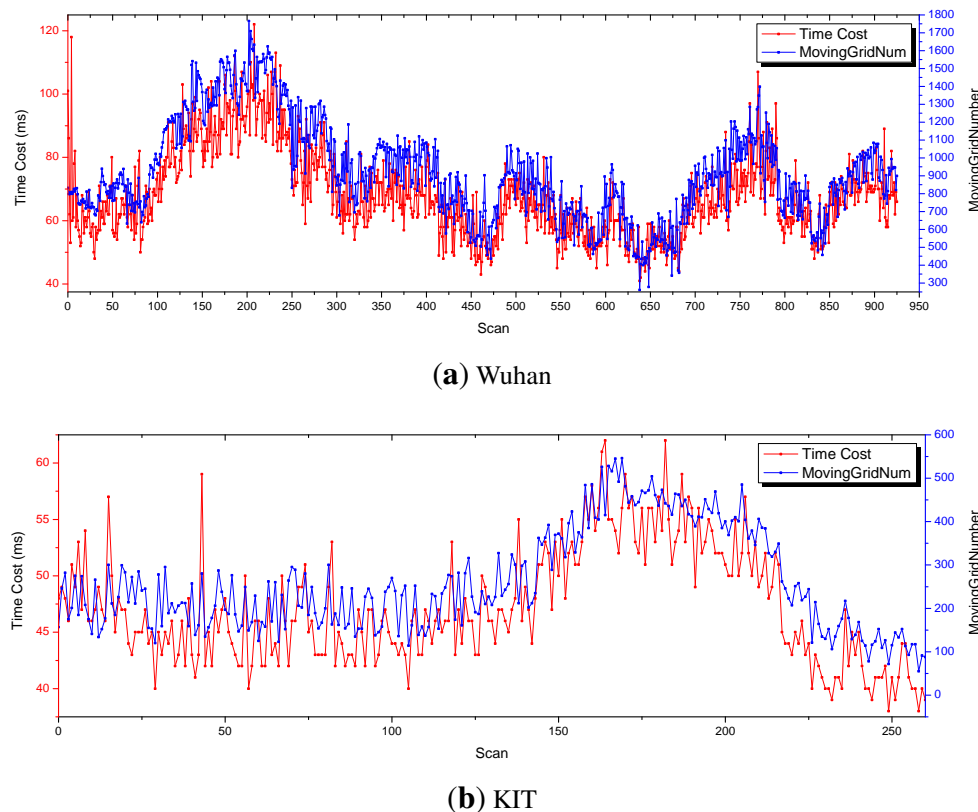
Figure 11. Time cost test.



(a) Time-cost of Wuhan data set



(b) Time-cost of KIT data set

Figure 12. The relationship between time-cost and the number of moving grids.

4.3. Time Cost

To operate in real time is one of the key requirements for sensing algorithms for use in active collision avoidance and intelligent driverless vehicle systems. The time cost of the proposed MFE algorithm is tested here.

Figure 11 shows the time cost for each of the three data sets with various longitudinal sensing distances (30 m, 40 m and 50 m, respectively). Obviously, as the sensing distance increases, so does the time cost increase. The average computation time for each of the three experimental data sets was less than 100 ms (the time interval of data output), so the proposed MFE is able to run in real time.

The relationship between time cost and the number of moving grids is also probed. In this experiment, the maximum longitudinal sensing distance is set as 40 m. As clearly shown in Figure 12, the time cost is proportional to the number of moving grids.

5. Conclusions

In this paper, we propose a motion field estimation method based on a 3D LiDAR and present both the theoretical framework and a practical solution. The proposed fast data association algorithm enables the algorithm to run in real time and the spatial smoothing algorithm, LMS mean smoothing, dramatically optimizes the motion field. Experimental results on several real urban data sets show that the proposed MFE method has excellent performance with high accuracy, good time consistency and spatial consistency. This proposed MFE method can enhance the environmental sensing ability of both

intelligent vehicles and active collision avoidance systems. In future work, we will seek to further improve the accuracy of motion state estimation and will research a method to globally optimize the motion field.

To conclude, the contribution of this paper is threefold: firstly, we provide a detailed method to construct the motion field from the raw 3D LiDAR data; secondly, we present a fast data association algorithm to enable the method to run in real time; thirdly, we propose a spatial-smoothing algorithm to optimize the motion field.

Acknowledgments

The authors would like to thank Christoph Stiller, Raquel Urtasun, Frank Moosmann, Andreas Geiger and Philip Lenz from Karlsruhe Institute of Technology for providing their data.

This work was jointly supported by the National Natural Science Foundation of China (Grant No. 41201483, No. 91120002, No. 61301277), Shenzhen Scientific Research and Development Funding Program (Grant No. ZDSY20121019111146499, No. JSGG20121026111056204), China Postdoctoral Science Foundation Funded Project (Grant No. 2012M521472) and Hubei Provincial Natural Science Foundation (No. 2013CFB299).

Author Contributions

The framework was proposed by Qingquan Li, and further development and implementation was realized by Liang Zhang. Qingzhou Mao and Qin Zou mainly studied some of the ideas. Pin Zhang mainly studied the actual experiment data collection and experiment process. Shaojun Feng and Washington Yotto Ochieng proof-read the paper and made some corrections.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Petrovskaya, A.; Thrun, S. *Model Based Vehicle Tracking for Autonomous Driving in Urban Environments*; The MIT Press: Zurich, Switzerland, 2008.
2. Shackleton, J.; vanVoorst, B.; Hesch, J. Tracking People with a 360-Degree Lidar. In Proceedings of the 2010 7th IEEE International Conference on Advanced Video and Signal Based Surveillance, Boston, MA, USA, 29 August–1 September 2010; pp. 420–426.
3. Kalyan, B.; Lee, K.W.; Wijesoma, W.S.; Moratuwage, D.; Patrikalakis, N.M. A Random Finite Set based Detection and Tracking Using 3D LIDAR in Dynamic Environments. In Proceedings of 2010 IEEE International Conference on Systems Man and Cybernetics (SMC), Istanbul, Turkey, 10–13 October 2010; pp. 2288–2292.
4. Liang, Z.; Qingquan, L.; Ming, L.; Qingzhou, M.; Nuchter, A. Multiple Vehicle-Like Target Tracking Based on the Velodyne Lidar. In Proceedings of the 8th IFAC Symposium on Intelligent Autonomous Vehicles, Gold Coast, Australia, 26–28 June 2013; Volume 8, pp. 126–131.

5. Moosmann, F.; Stiller, C. Joint Self-Localization and Tracking of Generic Objects in 3D Range Data. In Proceedings of 2013 IEEE International Conference on Robotics and Automation (ICRA), Karlsruhe, Germany, 6–10 May 2013; pp. 1146–1152.
6. Driessen, J.N.; Biemond, J. Motion Field Estimation for Complex Scenes. *Proc. SPIE* **1991**, *1605*, 511–521.
7. Dufaux, F.; Moscheni, F. Motion Estimation Techniques for Digital TV: A Review and a New Contribution. *Proc. IEEE* **1995**, *83*, 858–876.
8. Hosur, P.I.; Ma, K.K. Motion Vector Field Adaptive Fast Motion Estimation. In Proceedings of the 2nd International Conference on Information, Communications and Signal Processing (ICICS'99), Singapore, 7–10 December 1999; pp. 7–10.
9. Farnebäck, G. Very High Accuracy Velocity Estimation using Orientation Tensors, Parametric Motion, and Simultaneous Segmentation of the Motion Field. In Proceedings of the 8th IEEE International Conference on Computer Vision, Vancouver, BC, Canada, 7–14 July 2001; Volume I, pp. 171–177.
10. Franke, U.; Rabe, C.; Badino, H.; Gehrig, S.K. 6D-Vision: Fusion of Stereo and Motion for Robust Environment Perception. In Proceedings of the 27th DAGM Symposium, Vienna, Austria, 31 August–2 September 2005; pp. 216–223.
11. Rabe, C.; Franke, U.; Gehrig, S. Fast Detection of Moving Objects in Complex Scenarios. In Proceedings of the IEEE Intelligent Vehicles Symposium, Istanbul, Turkey, 13–15 June 2007; pp. 398–403.
12. Rabe, C.; Mifflin, T.; Wedel, A.; Franke, U. Dense, Robust, and Accurate Motion Field Estimation from Stereo Image Sequences in Real-Time. In Proceedings of the 11th European Conference on Computer Vision, Crete, Greece, 5–11 September 2010; Volume 6314, pp. 582–595.
13. Sellent, A.; Eisemann, M.; Goldlücke, B.; Cremers, D.; Magnor, M. Motion Field Estimation from Alternate Exposure Images. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)* **2011**, *33*, 1577–1589.
14. Xu, L.; Jia, J.; Matsushita, Y. Motion Detail Preserving Optical Flow Estimation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 1744–1757.
15. Hadfield, S. The Estimation and Use of 3D Information, for Natural Human Action Recognition. Ph.D. Thesis, University of Surrey, Surrey, UK, 2013.
16. Koller, J.; Ulmke, M. Multi Hypothesis Tracking of Ground Moving Targets. *GI Jahrestagung* **2005**, *68*, 307–311.
17. Chuang, K.-C.; Mori, S.; Chong, C.-Y. Evaluating a Multiple-Hypothesis Multitarget Tracking Algorithm. *IEEE Trans. Aerosp. Electron. Syst.* **1994**, *30*, 578–590.
18. Ryoo, M.S.; Aggarwal, J.K. Observe-and-explain: A New Approach for Multiple Hypotheses Tracking of Humans and Objects. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2008), Anchorage, AK, USA, 23–28 June 2008; pp. 1–8.
19. Buzzi, S.; Grossi, E.; Lops, M. Dynamic Programming Techniques for Sequential Detection and Tracking of Moving Targets in Radar Systems. *IEEE Trans. Aerosp. Electron. Syst.* **1996**, *32*, 1440–1451.

20. Boers, Y.; Driessen, H.; Torstensson, J.; Trieb, M.; Karlsson, R.; Gustafsson, F. A Track Before Detect Algorithm for Tracking Extended Targets. *IEE Proc.-Radar Sonar Navig.* **2006**, *153*, 345–351.
21. Tekinalp, S.; Alatan, A.A. Efficient Bayesian Track-Before-Detect. In Proceedings of 2006 IEEE International Conference on Image Processing, Atlanta, GA, USA, 8–11 October 2006; pp. 2793–2796.
22. Davey, S.J.; Rutten, M.G.; Cheung, B. A Comparison of Detection Performance for Several Track-before-Detect Algorithms. In Proceedings of 2008 11th International Conference on Information Fusion, Cologne, Germany, 30 June–3 July 2008; pp. 1–8.
23. Pertila, P.; Hamalainen, M. A Track Before Detect Approach for Sequential Bayesian Tracking of Multiple Speech Sources. In Proceedings of 2010 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), Dallas, TX, USA, 14–19 March 2010; pp. 4974–4977.
24. Guo, M.; Deng, X. An Improved Dynamic Programming Based Track-before-Detect Approach for Dim Target Detection. In *Communications, Signal Processing, and Systems*; Springer: New York, NY, USA, 2012; Volume 202, pp. 325–334.
25. Mazurek, P. Code Reordering Using Local Random Extraction and Insertion (LREI) Operator for GPGPU-Based Track-before-Detect Systems. *Soft Comput.* **2013**, *17*, 1095–1106.
26. Yi, W.; Morelande, M.R.; Kong, L.; Yang, J. An Efficient Multi-Frame Track-before-Detect Algorithm for Multi-Target Tracking. *J. Sel. Top. Signal Process.* **2013**, *7*, 421–434.
27. Velodyne LiDAR 2014. Available online: <http://velodynelidar.com/lidar/lidar.aspx> (accessed on 4 September 2013).
28. Lowe, D.G. Object Recognition from Local Scale-Invariant Features. In Proceedings of the 7th IEEE International Conference on Computer Vision, Kerkyra, Greece, 20–27 September 1999; Volume 2, pp. 1150–1157.
29. Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110.
30. Bertsekas, D.P. *Linear Network Optimization: Algorithms and Codes*; The MIT Press: Cambridge, MA, USA, 1991; pp. 1122–1127.
31. Jonker, R.; Volgenant, A.; Amsterdam. A Shortest Augmenting Path Algorithm for Dense and Sparse Linear Assignment Problems. *Computing* **1987**, *38*, 325–340.
32. Alparone, L.; Barni, M.; Bartolini, F.; Cappellini, V. Adaptively Weighted Vector-Median Filters for Motion-Fields Smoothing. In Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, Atlanta, GA, USA, 7–10 May 1996; pp. 2267–2270.
33. Guo, J.; Kim, J. Adaptive Motion Vector Smoothing for Improving Side Information in Distributed Video Coding. *J. Inf. Process. Syst.* **2011**, *7*, 103–110.
34. Frank, M.; Christoph, S. Velodyne SLAM. In Proceedings of the IEEE Intelligent Vehicles Symposium, Baden-Baden, Germany, 5–9 June 2011; pp. 393–398.