


## Article

# Real-Time HD Map Change Detection for Crowdsourcing Update Based on Mid-to-High-End Sensors

Pan Zhang <sup>1</sup>, Mingming Zhang <sup>2</sup> and Jingnan Liu <sup>1,\*</sup>

<sup>1</sup> School of Geodesy and Geomatics, Wuhan University, Wuhan 430072, China; panz@whu.edu.cn

<sup>2</sup> Gui Zhou Kuandeng Zhiyun Science and Technology Ltd., Beijing Branch, Beijing 100016, China; zhangmingming@kuandeng.com

\* Correspondence: jnliu@whu.edu.cn

**Abstract:** Continuous maintenance and real-time update of high-definition (HD) maps is a big challenge. With the development of autonomous driving, more and more vehicles are equipped with a variety of advanced sensors and a powerful computing platform. Based on mid-to-high-end sensors including an industry camera, a high-end Global Navigation Satellite System (GNSS)/Inertial Measurement Unit (IMU), and an onboard computing platform, a real-time HD map change detection method for crowdsourcing update is proposed in this paper. First, a mature commercial integrated navigation product is directly used to achieve a self-positioning accuracy of 20 cm on average. Second, an improved network based on BiSeNet is utilized for real-time semantic segmentation. It achieves the result of 83.9% IOU (Intersection over Union) on Nvidia Pegasus at 31 FPS. Third, a visual Simultaneous Localization and Mapping (SLAM) associated with pixel type information is performed to obtain the semantic point cloud data of features such as lane dividers, road markings, and other static objects. Finally, the semantic point cloud data is vectorized after denoising and clustering, and the results are matched with a pre-constructed HD map to confirm map elements that have not changed and generate new elements when appearing. The experiment conducted in Beijing shows that the method proposed is effective for crowdsourcing update of HD maps.

**Keywords:** HD map; crowdsourcing update; semantic segmentation; visual SLAM; autonomous driving



**Citation:** Zhang, P.; Zhang, M.; Liu, J. Real-Time HD Map Change Detection for Crowdsourcing Update Based on Mid-to-High-End Sensors. *Sensors* **2021**, *21*, 2477. <https://doi.org/10.3390/s21072477>

Academic Editor: Jari Nurmi

Received: 3 February 2021

Accepted: 30 March 2021

Published: 2 April 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Autonomous vehicles use various sensors to achieve different levels of autonomy (L1–L5, e.g., see [1]), such as cameras, Global Navigation Satellite System (GNSS), Radio Detection and Ranging (RADAR), Light Detection and Ranging (LIDAR). However, these sensors have a limited perception range, and they are very vulnerable to bad weather. To overcome the limitations, the pre-built digital map can be utilized to improve perception and robustness. Many autonomous vehicle prototypes rely on precise 3D maps [2,3], which are also called high-definition (HD) maps. An HD map is a precise map with rich lane-level information for autonomous driving. It can provide prior information robustly about the static environment in a range of more than 200 m ahead or around corners. The features in the map can be fused with the recognition results from camera/LIDAR to realize high accuracy localization of the vehicle [4].

Compared with the car navigation map, an HD map greatly improves the accuracy to a few centimeters level [5]. It also has richer and more detailed content, such as lane boundaries, lane centerlines, road markings on the ground, and guardrails on both sides of the road. Lane boundaries have many attributes in HD maps, such as type, color, and width. Therefore, HD maps reflect a more realistic and detailed real world, containing a lot of subtle changes. For example, the lane boundary is re-brushed, and arrows are added to the road surface.

At present, the production of HD maps requires professional data collection, that is professional surveying and mapping using the Mobile Mapping System (MMS) [6,7]. Then,

HD maps are constructed from the road images and 3D point cloud data. The entire data collection and production takes a long time. In addition, the professional survey fleet is very expensive to set up. All of these make it difficult to update the HD map in real-time.

Therefore, more and more researchers focus on crowdsourcing update of HD maps. In the future, driverless cars will be no different from professional survey cars as they are equipped with similar sensors. So, when they are driving, they will also be collecting data. The collected data of every car can be aggregated and then used to update HD maps. This is the concept of crowdsourcing updates of HD maps. Once the map update is completed in the cloud, the update package can be passed back to the vehicles. Then real-time HD map updates and services can be realized.

Ref [8] uses a single front-facing camera, a consumer-grade GNSS/IMU, a Qualcomm Snapdragon 820A SoC in the vehicle, and a backend mapping server to realize the crowdsourcing update of traffic signs and lane boundaries. Every single-journey perception data and triangulation outputs are shipped over a commercial LTE link to the backend mapping server. Thus, the amount of data transmitted by the network is very large. [9] proposed a generation method of new feature layers in the accuracy level of the HD map using the existing HD map and crowd-sourced information without additional costs. The generated new feature layer is uploaded to the map cloud by the mobile network. The amount of data transmitted is small, but the computing power requirements on the terminal are high. In addition, it focuses on the new feature types that have not appeared on the HD map.

Therefore, there are many different strategies based on different sensors and processing methods for the crowdsourcing update solution.

- First, what kind of sensors are there in the car, and what is the accuracy range of these sensors? For example, is the LIDAR included, and what is the accuracy of GNSS?
- Second, whether the full amount of sensing raw data is uploaded to the cloud, only some keyframe data is uploaded, or just the recognized results are uploaded.
- Third, is there an HD map on the end? If yes, whether the difference data between the HD map and real-time environment perception needs to be uploaded.

There are no very definite answers to these questions. However, there is no doubt that a reasonable sensor configuration and the corresponding processing method are very important for large-scale crowdsourcing update in order to ensure efficiency and reduce the amount of transmission data.

Currently, most social vehicles are not equipped with advanced sensors, only cameras for dashcam and low-precision positioning system for navigation. There is no strong perceptual computing power to process the original image. However, in recent years, mass-produced cars with automatic driving above L2 level, such as Audi A8 and BMW iNext, are not only equipped with cameras, millimeter-wave radar, and other sensors, but also equipped with a high-precision positioning system, HD maps, as well as chips or domain controllers with powerful real-time sensing computing power. Therefore, crowdsourcing map updating based on mid-high-end sensors is becoming more feasible and will be the trend in the future. In this paper, based on these mid-high-end sensors, vectorized real-time perception data is generated after real-time semantic segmentation, SLAM and other key technical processing. Then through the matching between the vectorized data and the pre-constructed HD map, map elements that have not changed are confirmed and new elements that are not on the map but in the real world are generated. This also means that real-time HD map change detection is realized for crowdsourcing update.

## 2. The Architecture

This paper proposes a real-time HD map change detection method in the vehicle terminal based on an industry camera, a high-end GNSS/IMU, and a high-performance onboard computing platform. Semantic SLAM (Simultaneous Localization and Mapping) technology is mainly used to obtain semantic point cloud data of lanes and other features based on an improved BiSeNet network [10]. Then, the semantic point data is vectorized

and matched with the HD map to detect differences. The proposed architecture is shown in Figure 1.

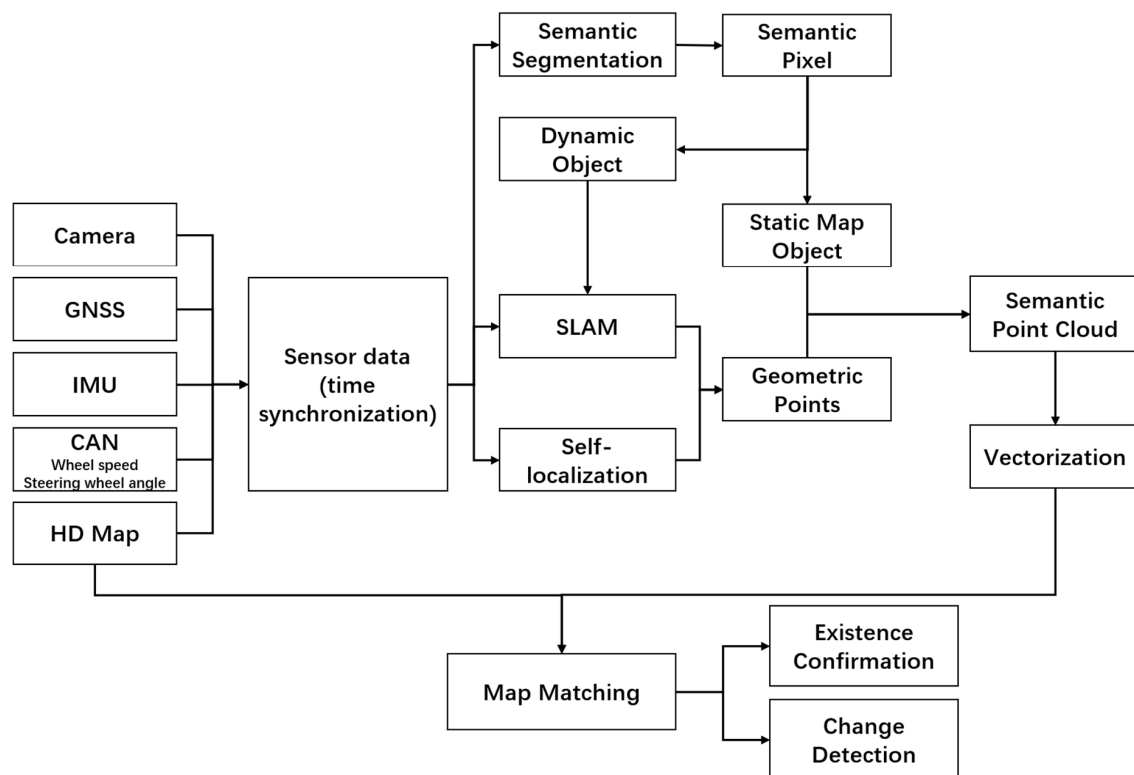


Figure 1. The architecture.

According to the architecture, the system comprises several modules as follows.

- **Localization.** Localization is one of the major subsystems of autonomous vehicles. Currently, the main sensors used for localization on vehicles are GNSS, IMU, cameras, and Controller Area Network Bus (CAN Bus) information such as wheel odometer. GNSS and IMU are commonly used to form an inertial navigation system. In this paper, NovAtel SPAN-IGM-A1 [11], a commercial integrated navigation product, is directly used to simulate the high-end positioning system on the vehicle. When using RTK (real-time kinematic) mode, it can provide a self-positioning accuracy of 20 cm on average. In this paper, we do not address this module.
- **Image semantic segmentation.** Semantic segmentation amounts to assign semantic labels to each pixel. With the development of deep learning, some networks could achieve good performance in semantic segmentation [12]. BiSeNet is one of the best real-time semantic segmentation networks in recent years. Thus, it is chosen to recognize the images collected by a single front-facing camera. In fact, this paper optimizes the BiSeNet network to improve recognition performance. NVIDIA DRIVE AGX Pegasus™ is utilized to simulate the onboard high-performance computing platform. Static features such as lane boundaries, road markings, traffic signs, and moving objects such as vehicles can be recognized with a relative high IOU (Intersection over Union). IOU is a commonly used measure for determining how accurate a proposed image segmentation is, compared to a ground-truth segmentation. Section 3 covers the image semantic segmentation in detail.
- **Semantic visual SLAM.** SLAM aims to self-localize a robot and estimate a model of its environment from sensory information [13]. The framework of the visual SLAM system is quite mature, which is generally composed of several essential parts such as front-end, back-end, and loop closure detection [14]. Some advanced SLAM

algorithms have already attained satisfactory performance, such as feature-based ORB-SLAM2 [15], direct method LSD-SLAM [16], and semi-direct visual odometer method (SVO) [17]. Vision SLAM can produce geometric maps composed of points or edges but without any associated meaning or semantic content [18]. As every pixel in the image has a known type after semantic segmentation, the usual SLAM maps can be enriched by associating the geometric estimation with object information. This process is called semantic SLAM [19]. In addition, the adaptability of visual SLAM to the dynamic scene is generally poor because of the limitations of the sparse image features. The constructed map often contains moving objects. Due to the influence of moving objects, there will be residual shadows of moving objects on the map. The semantic segmentation results can be used to remove these moving objects in the dynamic scenes to improve the quality of SLAM maps. Through the implementation of semantic visual SLAM technology, we can get semantic point cloud data with the spatial location of features such as lane boundaries, road markings. The details of semantic visual SLAM are described in Section 4.

- Vectorization and HD map matching. The semantic point cloud data is vectorized after denoising and clustering. The KD-tree and RANSAC algorithms are used. Then these vectorization results are matched with the local HD map to detect changes. The details of vectorization and HD map matching are given in Section 5.

### 3. Semantic Segmentation

NVIDIA DRIVE AGX Pegasus<sup>TM</sup> is used to simulate the vehicle-mounted high-performance computing platform. It achieves a 320 TOPS (Tera Operations Per Second) of supercomputer. Its next generation product will increase the computing power several times [20]. Therefore, it is foreseeable that the on-board computing power will continue to increase. This also means that more and more processing can be done on the end, such as images semantic segmentation, object detection, and so on.

BiSeNet [10] is a real-time semantic segmentation network proposed by Megvii Technology on ECCV2018. Using Res18 as a base model, the fast version of BiSeNet achieved the result of 74.8% Mean IOU (mIOU) on the CitiScapes verification dataset at 65.5 FPS. Our application scenarios require far less real-time performance of semantic segmentation than 60 fps, so we can improve the network to reduce the real-time performance, but increase the segmentation accuracy.

The original BiSeNet consists of two parts: Spatial Path (SP) and Context Path (CP). These two components are used to confront with the loss of spatial information and shrinkage of the receptive field. Spatial Path is designed to retain the spatial information of the original image. Context Path utilizes the lightweight model and global average pooling to quickly obtain a large receptive field. We have made targeted optimizations to these two parts. The improved network architecture is shown in Figure 2. The details of the improvement are as follows.

- (1) The original network uses the method of upsampling 8 times, 8 times, and 16 times in the last three output layers to directly restore the original size. It is modified to restore the image size by 4 times, 4 times, and 8 times through deconvolution. Finally, the original image size is restored by 2 times upsampling directly.
- (2) The classic attention idea is used in the original network, that is, average global pooling is utilized to obtain a sizeable receptive field. After our optimization, local attention and multi-scale attention are used to further improve the segmentation performance.

The modifications are shown in Figure 2, such as the abbreviation “deconv” for deconvolution. The key point is that we use deconvolution for restoring the original size of the image. The realization of deconvolution needs to obtain parameters through learning, so as to achieve higher accuracy. The original BiSeNet uses interpolation for upsampling directly. The advantage is that it is fast and does not require to obtain parameters. The disadvantage is that the accuracy is lower than deconvolution.

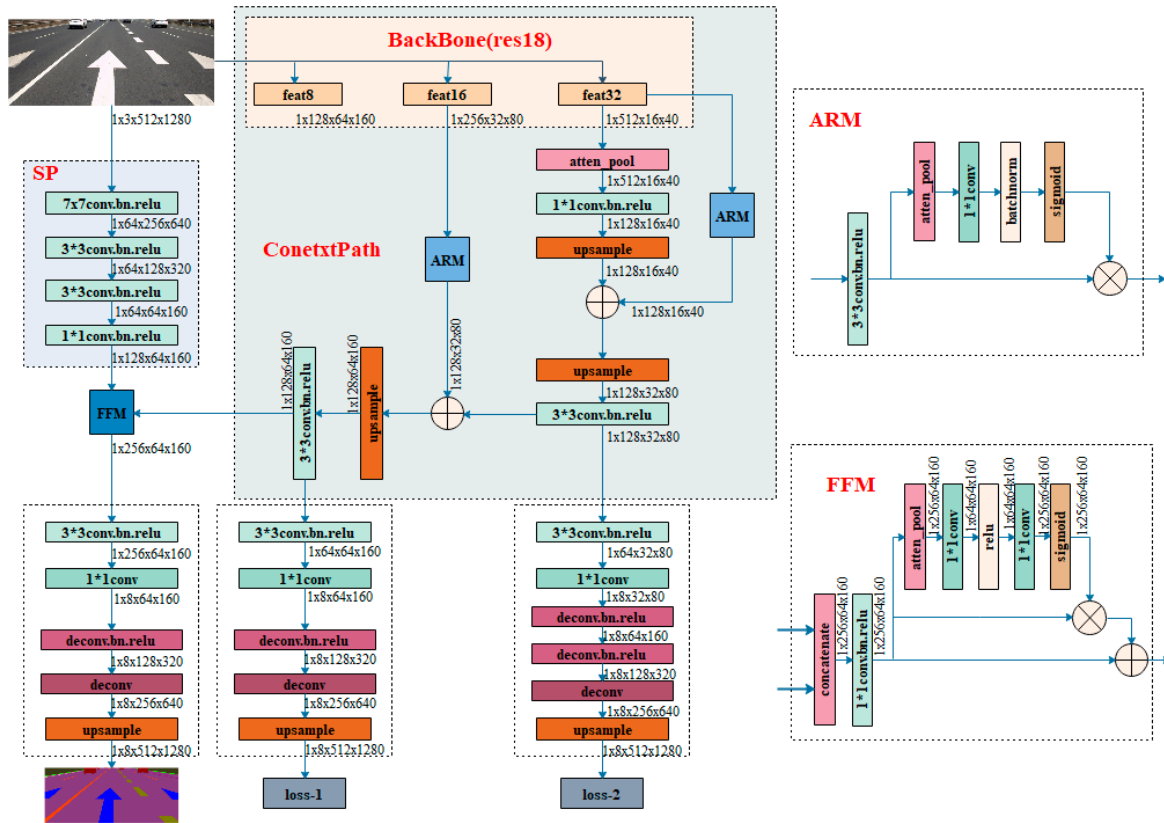


Figure 2. The improved network based on BiSeNet.

Regarding the loss function, the original BiSeNet uses Softmax loss and we adopt focal-loss [21]. Softmax loss is essentially a kind of cross-entropy loss function [22]. Focal-loss adds weight on the basis of cross-entropy and solves the problem of sample imbalance. So focal-loss performs better in accuracy.

The first step of focal-loss is the cross-entropy loss function for binary classification, defined as:

$$CE(p, y) = \begin{cases} -\log(p) & \text{if } y = 1 \\ -\log(1 - p) & \text{otherwise.} \end{cases} \quad (1)$$

In the above,  $y \in \{-1, 1\}$  specifies the ground-truth class and  $p \in [0, 1]$  indicates the model's estimated probability for the class with label  $y = 1$ . For simplicity, we define  $p_t$ :

$$p_t = \begin{cases} p & \text{if } y = 1 \\ 1 - p & \text{otherwise,} \end{cases} \quad (2)$$

Thus,

$$CE(p, y) = CE(p_t) = -\log(p_t) \quad (3)$$

Then in order to address class imbalance, a weighting factor  $\alpha \in [0, 1]$  is introduced:

$$\alpha_t = \begin{cases} \alpha & \text{if } y = 1 \\ 1 - \alpha & \text{otherwise,} \end{cases} \quad (4)$$

For simplicity, we define  $\alpha_t$  like we defined  $p_t$ .

$$CE(p_t) = -\alpha_t \log(p_t) \quad (5)$$

Then a modulating factor  $(1 - p_t)^\gamma$  is added to the cross-entropy loss for reducing the loss contribution from easily classified samples:  $\gamma \geq 0$ , which is a tunable focusing parameter.

$$m = (1 - p_t)^\gamma \quad (6)$$

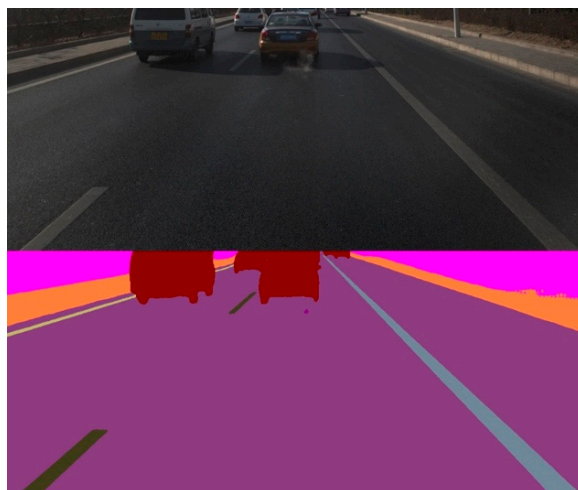
Thus, the focal loss is defined as:

$$FL(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t) \quad (7)$$

The data set for model training and verification comes from the image data collected by Kuandeng Technology. The original image resolution is  $2048 \times 2448$ , and there are more than 80 types of labeling categories. The data set has a total of 11,830 images, including 10,597 images in the training set and 1240 images in the verification set. The recognition results of the main categories are shown in the Table 1. The average IOU reaches 83.90%. While the average IOU of the original BiSeNet is 76.8%, which is very close to its results on the CitiScapes dataset. Thus, after our optimization, the accuracy has been improved. The IOU of objects on the road such as vehicles reaches 96.99%. The recognition performance of the stop line is relatively poor. It can be seen from the confusion matrix that the stop line is mainly misidentified as a road surface. In terms of inference speed, the improved BiSeNet's inference speed on Pegasus is about 31 fps, which is lower than the original BiSeNet but already meets our real-time requirements. An example of image semantic segmentation by the improved BiSeNet network is shown in Figure 3.

**Table 1.** Performance of improved BiSeNet network on Kuandeng verification dataset. IOU: Intersection over Union.

ID	Type	Recall	Pixel Accuracy	IOU
1	Traffic sign	94.40%	97.29%	91.98%
2	Pole	82.15%	90.05%	75.31%
3	Vehicles and other objects on the road	97.97%	98.97%	96.99%
4	Lane divider-white	92.42%	96.81%	89.69%
5	Lane-divider-yellow	77.32%	84.57%	67.76%
6	Speed bump	88.17%	93.42%	83.02%
7	Road surface	99.41%	98.89%	98.31%
8	Crosswalk	86.60%	97.79%	84.93%
9	Gore	94.90%	94.11%	89.57%
10	Text and symbol on the road	90.16%	93.40%	84.76%
11	Curb	86.45%	90.11%	78.96%
12	Others (Sky\Trees)	99.61%	99.37%	98.99%
13	Left road boundary	82.99%	91.94%	77.36%
14	Right road boundary	90.26%	93.14%	84.63%
15	Stop line	53.52%	94.82%	52.00%
16	Dedicated lane dividers	94.79%	92.55%	88.07%
	Average	88.20%	94.20%	83.90%



**Figure 3.** An example of image semantic segmentation.



#### 4. Semantic SLAM

After semantic segmentation, each pixel in the image is labeled with semantic tags. Then the dynamic vehicles and other outliers in the image can be filtered in the process of tracking in the dynamic environment. The meaning of the static features can be associated with points after visual SLAM mapping.

The front-end of SLAM is also called visual odometer (VO), which tracks the camera's position and pose through the geometric relationship between multiple views. Because semi-direct VO (SVO) can combine the success-factors of feature-based methods (tracking many features) with the accuracy and speed of direct methods, it is adopted as the front end of our Visual SLAM system. As a mature and open-source method, the detailed implementation of SVO can be found in [17].

With the continuous increase of images and the continuous operation of the SLAM system, the observation error of each frame will accumulate to the next. Thus, the measurement error will continue to accumulate. Therefore, it is very important for the SLAM system to optimize trajectory on the back-end. BA (bundle adjustment) [23,24] based on graph optimization is commonly used for global optimization. The open-source graph optimization library g2o (general graph optimization) that contains the implementation of BA is utilized in this process.

When we know the camera pose from the motion estimation, the depth at a single pixel can be estimated from multiple observations by means of a recursive Bayesian depth filter [17]. From the pixel with highest correlation in the epipolar line, the depth measurement is triangulated to update the depth filter. For forward motions, it is beneficial to update the depth filters with the previous frame. While in the older version of SVO [25], the depth filter is only updated with newer frames, which works well for down-looking cameras in micro aerial vehicle applications. In fact, whether it is SVO or feature-based SLAM, such as OBR-SLAM2, triangulation is the most basic depth estimation method. Through the triangulation method, all the keyframe points can be transformed into a unified coordinate system through the corresponding perspective transformation matrix, so as to generate the point cloud map.

#### 5. Vectorization and Matching with the HD Map

The vectorization of semantic point cloud data consists of three steps as shown in Figure 4. The first step is denoising. As the pixel accuracy of semantic segmentation is not 100%, the noise generated needs to be filtered out. After SLAM processing, the semantic point cloud is attached to the spatial position. Thus, the Euclidean distance measure can be used for KD-Tree construction to find  $n$  points near one point and then discard the points with long distance [26]. The second is clustering. The Euclidean cluster extraction algorithm is utilized to cluster the denoised point cloud data for every object. Some examples of clustering are shown in Figure 5. We can see that the performance is acceptable. Finally, vectorization is implemented. The minimum bounding box is calculated for the surface element as its geometry. An RANSAC algorithm [27] is used for curve fitting for line elements. These algorithms are not the focus of this article, so they are not described in detail. In the process of implementation, an open-source Point Cloud Library (PCL) is used directly, which provides a lot of general point-cloud-related algorithms and efficient data structures.

After denoising, clustering, and vectorization, the vectorized results are generated. For lane dividers, line-to-line matching is needed. The matching degree is normally equivalent to the similarity calculated through distance and angle. For polygon elements like arrows, the matching degree is evaluated by the proportion of the area covered. The matching between the vectorized polygon and the HD map is illustrated by an experiment conducted on the Fifth Ring Road in Beijing.

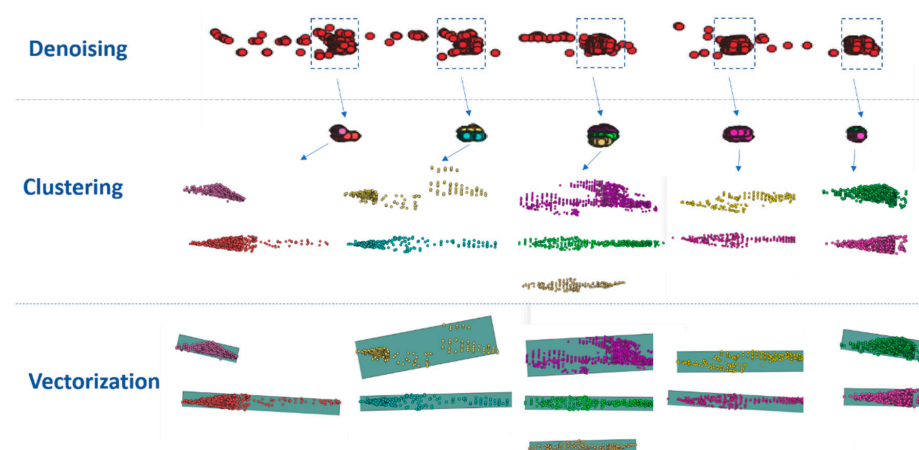


Figure 4. Vectorization of semantic point cloud data (taking arrows as an example).

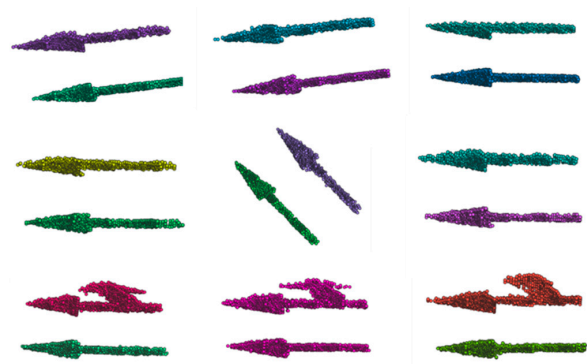


Figure 5. Examples of clustering of semantic point cloud.

As shown in Figures 6 and 7, the green point group is the semantic point cloud data after semantic visual SLAM. The green rectangle represents the bounding box of an arrow in the pre-constructed HD map. When the car is moving, the semantic point cloud continues to grow. The vectorized results extracted from the semantic point cloud can be used to either confirm the existence of map features or generate new ones when new map features appear. In order to do the experiment, some arrows were deleted from the pre-constructed HD map.

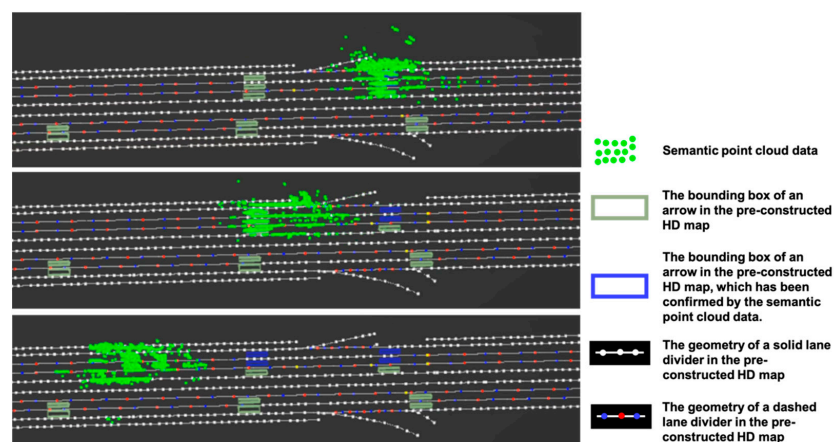


Figure 6. Confirmation of existence for map elements.



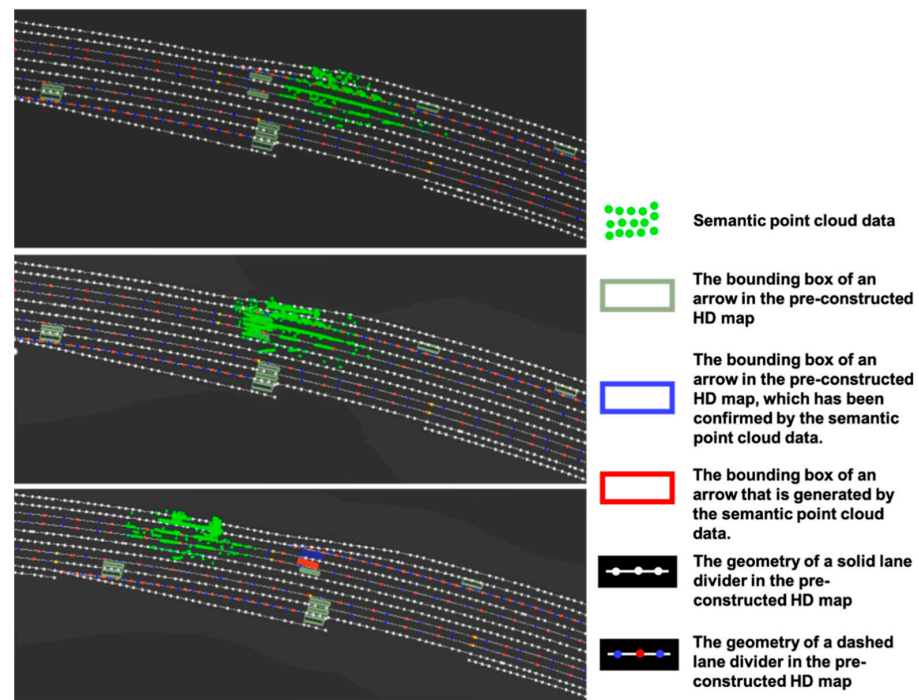


Figure 7. Generation of new map features from semantic point cloud.

In Figure 6, when the vehicle ran through a lane, the semantic point cloud is used to confirm the existence of arrows on the ground. It is indicated by a change of color from green to blue as shown in the middle and lower parts of Figure 6.

In Figure 7, the semantic point cloud is used to add the missing arrows, which is indicated by the new addition of red color as shown in the lower part of Figure 7.

Obviously, there is a difference in precision between the extracted features from semantic point cloud data and the map features. In order to find the corresponding map features, the matching degree between these two should be calculated. As shown in Figure 8, the green rectangle represents the bounding box of a polygon map element (e.g., an arrow), and the red represents the extracted element. The yellow part of their overlap is the area that is really matched.

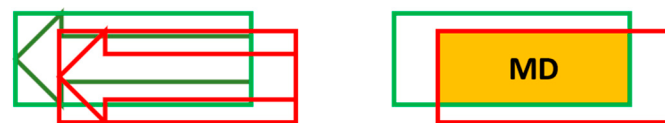


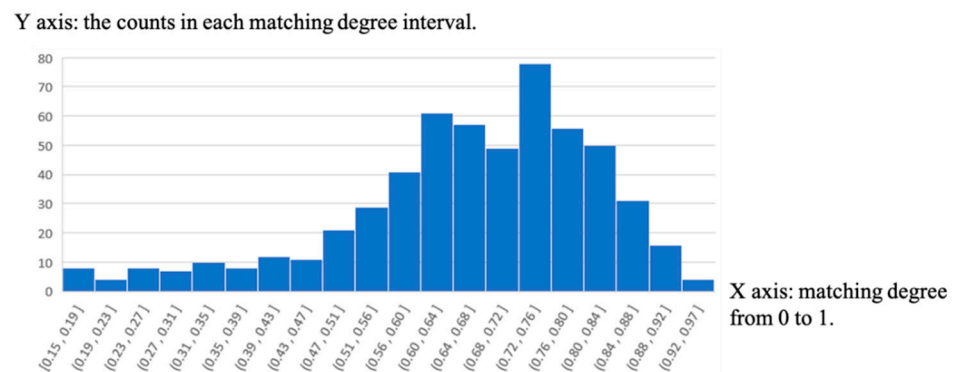
Figure 8. The case that bounding boxes of the map element and extracted feature cover each other.

The definition and calculation formula of the matching degree is as follows. That is, the matching degree is equal to the area of the overlapping divided by the area of the map element bounding box.

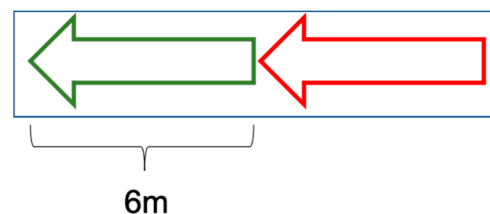
$$MD = \frac{\text{Area(Overlap)}}{\text{Area(Map)}} \quad (8)$$

The matching degree can reflect the accuracy of the semantic point cloud data. The statistical distribution of the matching degree is shown in Figure 9. The X axis represents the matching degree. The matching degrees from 0 to 1 are divided into intervals according to a fixed range 0.05. The Y axis represents the counts in each matching degree interval. As we can see, no matching degree is between 0 and 10%. Actually, the minimum matching degree is 14.6% if matching is successful. This can indicate that features that are not matched should not be included in the map data. In other words, the case shown in Figure 10 is

unlikely to happen. By the way, the length of arrows is 6 m. This conclusion is crucial to the application of crowdsourcing data.



**Figure 9.** Statistical results of matching degree.



**Figure 10.** The case that bounding boxes of the map element and extracted feature don't cover each other.

Regarding the data volume of the semantic point cloud, the average data volume per second is 1126 KB and the average data volume per kilometer is 56,704 KB. After vectorization and matching with the HD map, the difference data is identified, which has a much smaller amount of data for transmission. The amount of the difference data depends on the changes between the old HD map and the real-time reality. The more changes, the greater the amount of the different data. In our experiment, the average amount of the different data is 126 KB per kilometer, which is much less than the amount of the whole data.

In terms of accuracy and the amount of transmitted data, the results show that the sensors and methods proposed in this paper are effective for crowdsourcing update.

## 6. Conclusions

With the unlimited range of environment perception and other advantages, the HD map is widely considered to be an essential part of autonomous driving. However, due to the complex and expensive data collection and production of HD maps, continuous maintenance and real-time updates have become a huge challenge. With the development and maturity of autonomous driving technology, more and more vehicles are equipped with a variety of advanced sensors and a powerful computing platform. Therefore, the crowdsourcing update of HD maps based on mid-to-high-end sensors is becoming more feasible.

In this context, this article uses a mature commercial GNSS/IMU integrated navigation device, an industrial camera, and NVIDIA Pegasus with GPU (Graphics Processing Unit) to launch the research. The main method is to perform real-time semantic segmentation of images based on the improved BiSeNet network and then fuse the results with visual SLAM to obtain semantic point cloud data. After denoising, clustering, and vectorization, the vectorized results are extracted from the semantic point cloud data and then matched with a pre-constructed HD map. The map elements that have not changed can be confirmed and the elements that have changes can be detected. In the experiment, the existence of arrows in the HD map is confirmed and new arrows are generated. In summary, real-time

HD map change detection is realized and validated, which also demonstrates the feasibility and significant value of crowdsourcing update for HD maps.

**Author Contributions:** Conceptualization, P.Z. and J.L.; methodology, P.Z.; software, P.Z. and M.Z.; validation, P.Z. and M.Z.; formal analysis, P.Z.; investigation, P.Z. and M.Z.; resources, P.Z.; data curation, P.Z. and M.Z.; writing—original draft preparation, P.Z.; writing—review and editing, J.L.; visualization, P.Z. and M.Z.; supervision, J.L.; project administration, P.Z.; funding acquisition, J.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Strategy Research Project on “Beidou+5G” Technology and Industry Development in 2020 under Chinese Engineering Technology Development Strategy Hubei Research Institute, grant number HB2020B13.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Publicly available datasets were analyzed in this study. This data can be found here: <https://www.cityscapes-dataset.com> (accessed on 1 April 2021).

**Acknowledgments:** Many thanks to Gui Zhou Kuandeng Zhiyun Science and Technology Ltd. Beijing Branch for providing data and technical support.

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

## References

1. SAE Standard J3016\_201806: Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles. Available online: [https://saemobilus.sae.org/content/J3016\\_201806](https://saemobilus.sae.org/content/J3016_201806) (accessed on 1 April 2021).
2. Levinson, J.; Askeland, J.; Becker, J.; Dolson, J.; Held, D.; Kammel, S.; Kolter, J.Z.; Langer, D.; Pink, O.; Pratt, V.; et al. Towards fully autonomous driving: Systems and algorithms. In Proceedings of the 2011 IEEE Intelligent Vehicles Symposium (IV), Baden, Germany, 5–9 June 2011; pp. 163–168.
3. Ziegler, J.; Bender, P.; Schreiber, M.; Lategahn, H.; Strauss, T.; Stiller, C.; Dang, T.; Franke, U.; Appenrodt, N.; Keller, C.G.; et al. Making Bertha Drive—An Autonomous Journey on a Historic Route. *IEEE Intell. Transp. Syst. Mag.* **2014**, *6*, 8–20. [CrossRef]
4. Schreiber, M.; Knoppel, C.; Franke, U. LaneLoc: Lane marking based localization using highly accurate maps. In Proceedings of the 2013 IEEE Intelligent Vehicles Symposium (IV), Gold Coast, Australia, 23–26 June 2013; pp. 449–454.
5. Strijbosch, W. Safe Autonomous Driving with High-definition Maps. *ATZ Worldw.* **2018**, *120*, 28–33. [CrossRef]
6. Joshi, A.; James, M.R. Generation of Accurate Lane-Level Maps from Coarse Prior Maps and Lidar. *IEEE Intell. Transp. Syst. Mag.* **2015**, *7*, 19–29. [CrossRef]
7. Chen, A.; Ramanandan, A.; Farrell, J.A. High-precision lane-level road map building for vehicle navigation. *IEEE/ION Position Locat. Navig. Symp.* **2010**, 1035–1042. [CrossRef]
8. Dabeer, O.; Ding, W.; Gowaiker, R.; Grzechnik, S.K.; Lakshman, M.J.; Lee, S.; Reitmayr, G.; Sharma, A.; Somasundaram, K.; Sukhvasi, R.T.; et al. An end-to-end system for crowdsourced 3D maps for autonomous vehicles: The mapping component. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 634–641.
9. 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]
10. Yu, C.; Wang, J.; Peng, C.; Gao, C.; Yu, G.; Sang, N. BiSeNet: Bilateral Segmentation Network for Real-Time Semantic Segmentation. In Proceedings of the Formal Concept Analysis, Darmstadt, Germany, 27 December 2018; pp. 334–349.
11. SPAN-IGM-A1. Available online: <https://novatel.com/support/span-gnss-inertial-navigation-systems/span-combined-systems/span-igm-a1> (accessed on 10 January 2021).
12. Badrinarayanan, V.; Kendall, A.; Cipolla, R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [CrossRef] [PubMed]
13. Durrant-Whyte, H.; Bailey, T. Simultaneous localisation and mapping (SLAM): Part I the essential algorithms. *Robot. Autom. Mag.* **2006**, *13*, 99–110. [CrossRef]
14. Yu, C.; Liu, Z.; Liu, X.-J.; Xie, F.; Yang, Y.; Wei, Q.; Fei, Q. DS-SLAM: A Semantic Visual SLAM towards Dynamic Environments. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 1168–1174.
15. Mur-Artal, R.; Tardos, J.D. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Trans. Robot.* **2017**, *33*, 1255–1262. [CrossRef]
16. Engel, J.; Schops, T.; Cremers, D. LSD-SLAM: Large-scaledirect monocular SLAM. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 834–849.

17. Forster, C.; Zhang, Z.; Gassner, M.; Werlberger, M.; Scaramuzza, D. SVO: Semidirect Visual Odometry for Monocular and Multicamera Systems. *IEEE Trans. Robot.* **2016**, *33*, 249–265. [[CrossRef](#)]
18. Civera, J.; Gálvez-López, D.; Riazuelo, L.; Tardós, J.D.; Montiel, J.M.M. Towards semantic SLAM using a monocular camera. In Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, CA, USA, 25–30 September 2011; pp. 1277–1284.
19. McCormac, J.; Handa, A.; Davison, A.; Leutenegger, S. SemanticFusion: Dense 3D semantic mapping with convolutional neural networks. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 4628–4635.
20. NVIDIA Introduces DRIVE AGX Orin—Advanced, Software-Defined Platform for Autonomous Machines. Available online: <https://nvidianews.nvidia.com/news/nvidia-introduces-drive-agx-orin-advanced-software-defined-platform-for-autonomous-machines> (accessed on 17 December 2019).
21. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollar, P. Focal Loss for Dense Object Detection. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
22. De Boer, P.-T.; Kroese, D.P.; Mannor, S.; Rubinstein, R.Y. A Tutorial on the Cross-Entropy Method. *Ann. Oper. Res.* **2005**, *134*, 19–67. [[CrossRef](#)]
23. Triggs, B.; McLauchlan, P.F.; Hartley, R.I.; FitzGibbon, A.W. Bundle Adjustment—A Modern Synthesis. In *International Workshop on Vision Algorithms*; Springer: Berlin/Heidelberg, Germany, 1999; pp. 298–372.
24. Konolige, K.; Agrawal, M. FrameSLAM: From Bundle Adjustment to Real-Time Visual Mapping. *IEEE Trans. Robot.* **2008**, *24*, 1066–1077. [[CrossRef](#)]
25. Forster, C.; Pizzoli, M.; Scaramuzza, D. SVO: Fast semi-direct monocular visual odometry. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 15–22.
26. Greenspan, M.; Yurick, M. Approximate K-D tree search for efficient ICP. In Proceedings of the Fourth International Conference on 3-D Digital Imaging and Modeling, Banff, AB, Canada, 6–10 October 2003.
27. Chum, O.; Matas, J. Optimal Randomized RANSAC. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *30*, 1472–1482. [[CrossRef](#)] [[PubMed](#)]