IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received December 27, 2019, accepted January 22, 2020, date of publication February 3, 2020, date of current version February 10, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2971059

Incomplete Road Information Imputation Using Parallel Interpolation to Enhance the Safety of Autonomous Driving

KAIFENG GAO, BOWEN WANG, LEI XIAO, AND GANG MEI, (Member, IEEE)
School of Engineering and Technology, China University of Geosciences (Beijing), Beijing 100083, China

Corresponding author: Gang Mei (gang.mei@cugb.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 11602235 and Grant 41772326, and in part by the Fundamental Research Funds for China Central Universities under Grant 2652018091, Grant 2652018107, and Grant 2652018109.

ABSTRACT Autonomous driving is the main application of Internet of Things (IoT) technology in the field of intelligent transportation. In autonomous driving, self-driving cars will avoid changing lanes in a short distance. When the self-driving car executes the follow-up instruction, the road smoothness in front of the car will affect the driving safety and comfort of the car. The real-time acquisition of road information in front of driving will help the self-driving car adjust driving behavior. However, other vehicles on the road will lead to the failure of Light Detection And Ranging (LiDAR) detectors to obtain complete road point cloud data. The incomplete road point cloud data need to be imputated to avoid potential misjudgements of the road conditions. Currently, little research work specifically focuses on imputating the incomplete road point cloud data that are caused by obstacle vehicles. In this paper, we propose a fast method to imputate the incomplete road point cloud data using a Graphics Processing Unit (GPU)-based parallel Inverse Distance Weighted (IDW) interpolation algorithm to enhance the safety of autonomous driving. To evaluate the performance of the proposed method, two groups of experiments are conducted. The experimental results indicate the following: (1) the known point cloud data within 5 meters around the obstacle vehicle are sufficient to guarantee the imputation accuracy; (2) when the weight parameter of the IDW interpolation is 4, the efficiency and accuracy of the imputation can be optimally balanced; and (3) it takes approximately 0.6 seconds to imputate the incomplete dataset consisting of 15 million points, while the imputation error is approximately 5 millimeters. The proposed method is capable of efficiently and effectively imputating the incomplete road point cloud data that are induced by obstacle vehicles and outperforms other interpolation algorithms and machine learning algorithms.

INDEX TERMS IoT, self-driving car, path planning, data imputation, interpolation algorithm, GPU.

I. INTRODUCTION

Internet of Things (IoT) is one of the most exciting emerging technologies. IoT combines various information sensing devices with the Internet to exchange and communicate information, thus realizing intelligent identification, location, tracking, monitoring and management of items [1]–[3]. Currently, the application of the IoT involves fields as intelligent transportation [4], [5], intelligent medical treatment [6]–[8], public safety [9]–[11], etc. It effectively promotes the intelligent development of these fields, and makes the limited resources allocated more reasonably.

The associate editor coordinating the review of this manuscript and approving it for publication was Antonino Orsino.

In intelligent transportation, automatic driving vehicle is one of the most important solutions to realizing traffic intelligence. Recently, autonomous driving has received significant attention from scientific research institutions and automobile manufacturing enterprises. It is one of the most interesting scientific and technological research topics, and it is also a potential solution to enhancing driving safety, alleviating traffic congestion [12]–[14], and reducing air pollution [15]–[18].

Safety is an essential requirement for self-driving cars [19]–[21]. Environmental perception is the precondition of high-level intelligent behaviour such as obstacle avoidance and path planning for self-driving cars. One of the most important tasks in environmental perception is to collect and

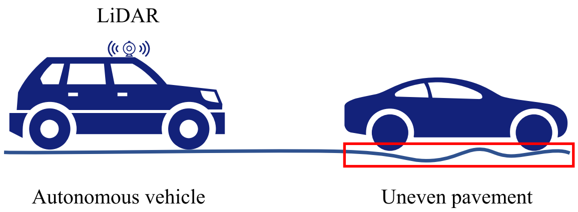


FIGURE 1. Self-driving car executes follow command on uneven pavement.

identify road information. The timely acquisition of road information during driving can provide important information for real-time path planning to ensure vehicle safety. Therefore, the IoT technology is an inevitable problem in the development of autonomous vehicles.

Effective road detection is challenging for self-driving cars [22]. In the process of autonomous driving, road recognition depends on GPS, sensors, and high-resolution digital map guidance [23]. A self-driving car can determine its orientation using GPS coordinates on a high-definition digital map and combining the environmental data that are obtained by the digital cameras and LiDAR sensors around the vehicle to identify the current road conditions.

To obtain timely, accurate, and reliable road information for decision-making, various cameras and sensors have been equipped onto self-driving cars to acquire the road data. The first is the vehicle vision system [24], [25]. An image is acquired by image acquisition devices, and then converted into a digital signal according to the pixel distribution, brightness, and colour. Finally, the required information is recognized by extracting the characteristics of the digital signals. The second is LiDAR [26], [27], which can continuously collect point cloud data to assess the environment around the vehicle during the driving process [28]–[30]. The laser beams in a LiDAR repetitively scan the surrounding environment, and produce the road point cloud data. In addition, ultrasonic sensors and infrared sensors are also used in environmental sensing.

In theory, for autonomous vehicles, the more important work is to detect the drivable area and avoid the surface that is occupied by the obstacles. However, in practice, the autonomous vehicle may try to avoid lane change in a short distance, because the planning control algorithm of the autonomous vehicle needs more lane change space than the normal driver needs for safety reasons [31]. In autonomous driving, when there are normal driving vehicles in front and around, a decision of behavioral decision-making is likely to be a follow-up order; see Figure 1. In this case, the flatness of the road ahead of the autonomous vehicle becomes an important factor that restricts the safety and stability of the autonomous vehicle [32], [33].

The autonomous vehicles realize the environment perception through the high-precision map combined with LiDAR and other sensors. But the update cycle of the high-precision map is approximately once a week, if there are new potholes or bumps on the road, the autonomous vehicles can only obtain the road information through real-time detection. In addition, the real-time detection of the road is more important when there is no high-precision map on the strange road [34]. When the pothole road is occupied by other vehicles, the radar detection results will be affected, which is not conducive to the safety of automatic driving [35]. Therefore, it is very important to quickly supplement the road information blocked by the vehicles in front to enhance the safety of autonomous vehicles.

Due to the fast decision-making requirements of autonomous vehicles, the rapid identification and imputation of incomplete road information are critical to ensure driving safety. Most importantly, both the efficiency and accuracy of the incomplete road information imputation need to be seriously guaranteed.

However, to the best of the authors' knowledge, little work specifically focuses on imputating the incomplete road information that is induced by obstacles such as vehicles. Most studies estimate the missing values in time-series data using various machine learning algorithms and mathematical algorithms such as the Radial Basis Function (RBF) interpolation, the Support Vector Machine (SVM) regression, and the k-Nearest Neighbours (kNN) estimation [36]–[39].

In this paper, we propose a fast method to imputate the incomplete road point cloud data using a GPU-based parallel IDW interpolation algorithm to enhance the safety of autonomous driving. There are two main steps in the proposed method. First, the known point cloud data within a certain distance around the obstacle vehicle are optimally identified to reduce the number of scattered points for imputation. Second, a simple and efficient GPU-based parallel IDW interpolation algorithm is employed to imputate the incomplete road point cloud data. To evaluate the performance of the proposed method, two groups of experiments are conducted.

The main contributions of this paper can be summarized as follows.

(1) We propose a fast and effective method to imputate the incomplete road point cloud data that are induced by obstacle vehicles to enhance the safety of autonomous driving.

(2) We employ a simple and efficient GPU-based paralell IDW interpolation algorithm to estimate the missing points in the incomplete road point cloud data.

The rest of the paper is organized as follows. Section 2 describes the proposed method in detail. Section 3 presents two benchmark experiments and analyzes the experimental results. Section 4 discusses and compares the proposed method with other relevant methods. Finally, Section 5 draws several conclusions.

II. METHOD

A. ESSENTIAL IDEAS OF INCOMPLETE ROAD INFORMATION IMPUTATION

A self-driving car needs to perceive the road information in the driving process, as shown in Figure 2. The complete road scene is crucial for real-time path planning. To reconstruct the three-dimensional real scene of the road, the

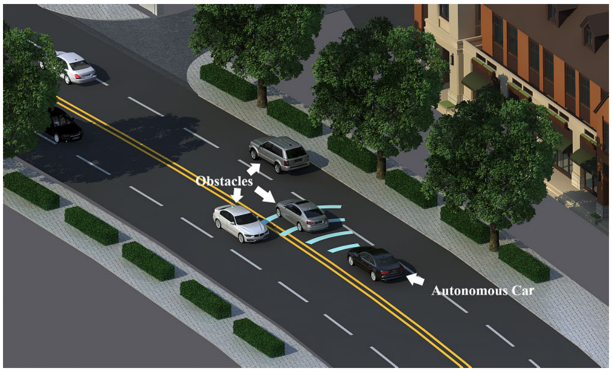


FIGURE 2. Driving scene of self-driving cars.

self-driving car usually scans the road in real-time. After scanning, the coordinates of the scattered points on the road surface are obtained, and the road scene can be reconstructed using the point cloud data.

However, if there are other vehicles or obstacles in the direction of driving, their presence will affect the collection of the road point clouds by LiDAR, resulting in the lack of road information in the area where the obstacles are located. To address this problem, we employ the GPU-accelerated parallel interpolation algorithm to fast imputate the missing road point cloud data based on the known road information around the obstacle, and provide the complete road information for path planning for the self-driving car in real time. This is shown in Figure 3. The process of imputating incomplete road information is illustrated in Figure 4.

First, we analyse the three-dimensional point cloud data that are obtained by LiDAR scanning. If there is a lack of road point cloud data, then it needs to estimate the missing road point cloud data for real-time path planning to ensure the safety of self-driving cars.

Before the interpolation of the missing point data, we need to determine the range and spatial location of the missing region. It is considered that if all point cloud data in the scanning area of LiDAR are used for the interpolation calculation, it will consume more time and thus affect the decision-making efficiency. Therefore, we use the known point cloud data set for the interpolation calculation, which is centred on the missing data area and extended to a certain range. Different extended ranges may lead to quite different computational efficiencies and accuracies, and thus, it needs to first determine the suitable parameters of the expansion through many experiments with different extended ranges.

After determining the known point cloud data set for interpolation, we interpolate the missing area point cloud data using the IDW interpolation method based on GPU parallel acceleration. It should be noted that the weight parameter of the IDW interpolation has a great impact on the interpolation accuracy. Therefore, the most suitable weight parameter of the IDW for imputating the missing point cloud data of roads can be found by calculating the different weight values.

To evaluate the efficiency and accuracy of the proposed method, we record all the times of the whole experiment and test process, calculate the interpolation results and compare the height of the random point set to determine the Root Mean Squared Error (RMSE). We use the computational time to evaluate the interpolation efficiency and the RMSE to evaluate the interpolation accuracy. The RMSE is defined via Eq.(1).

$$RMSE = \sqrt{\frac{\sum d_i^2}{n}} \tag{1}$$

where d_i is a set of deviations that are between the measured and true values, and n is the number of calculations.

B. GPU-ACCELERATED PARALLEL IDW INTERPOLATION ALGORITHM

IDW interpolation is a commonly used interpolation method, which has a simple form and is only related to the spatial distance. The IDW interpolation method mainly depends on the weight of the inverse distance. Weight parameters can be used to adjust the influence of known points based on the distance between the interpolation points and known points. The greater the weight of the inverse distance is, the greater the influence of the adjacent points on the interpolation points. Assuming that there are n known points, the weighting function is shown in Eq.(2).

$$\omega_i(x) = \frac{d(x, x_i)^{-p}}{\sum_{i=1}^{n} d(x, x_i)^{-p}} \tag{2}$$

where x is the position of the interpolation point, x_i is the position of the known point, d is the distance between the two points, and p is the inverse distance weight.

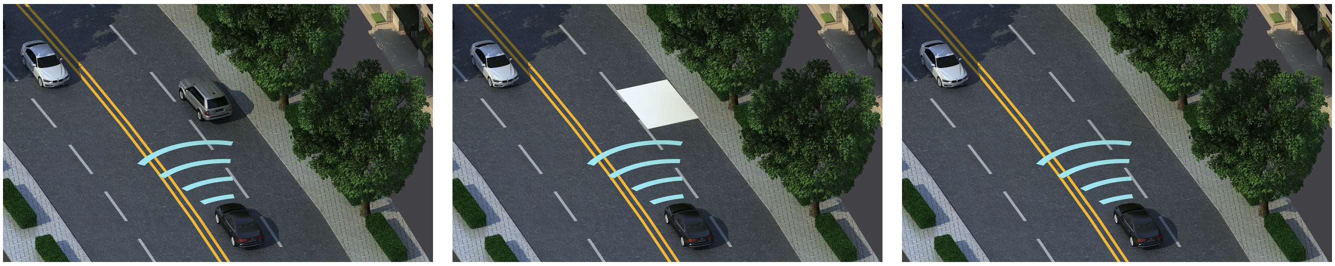
After calculating the weighting function for each known point to the interpolation point, the value of the interpolation point is calculated via Eq.(3).

$$Z(x) = \sum_{i=1}^{n} \omega_i(x) z_i \tag{3}$$

where z_i is the value of the i^{th} known point.

In GPU computing, data layouts may strongly affect the computing efficiency. Therefore, one of the key issues in developing efficient GPU implementations is to design an appropriate data layout when manipulating multi-valued data.Typically, there are two main data layouts in GPU computing, the Array of Structures (AoS) and the Structure of Arrays (SoA), which are shown in Figure 5. In this paper, we use the SoA data layout. Because there is no data staggering, organizing data according to the SoA layout can usually make full use of the memory bandwidth [40]. In addition, when high-frequency data access is required, the SoA is generally more efficient than the AoS, because the continuous storage of frequently accessed data will significantly improve access speed.

We use the optimized *tiled* version to implement the IDW interpolation [40]–[42]. In this version, the coordinates of the data points are first transferred from the global memory to the



(a) Initial road environment (b) Blind area of LiDAR scanning (c) After imputation

FIGURE 3. Illustration of incomplete road surface imputation.

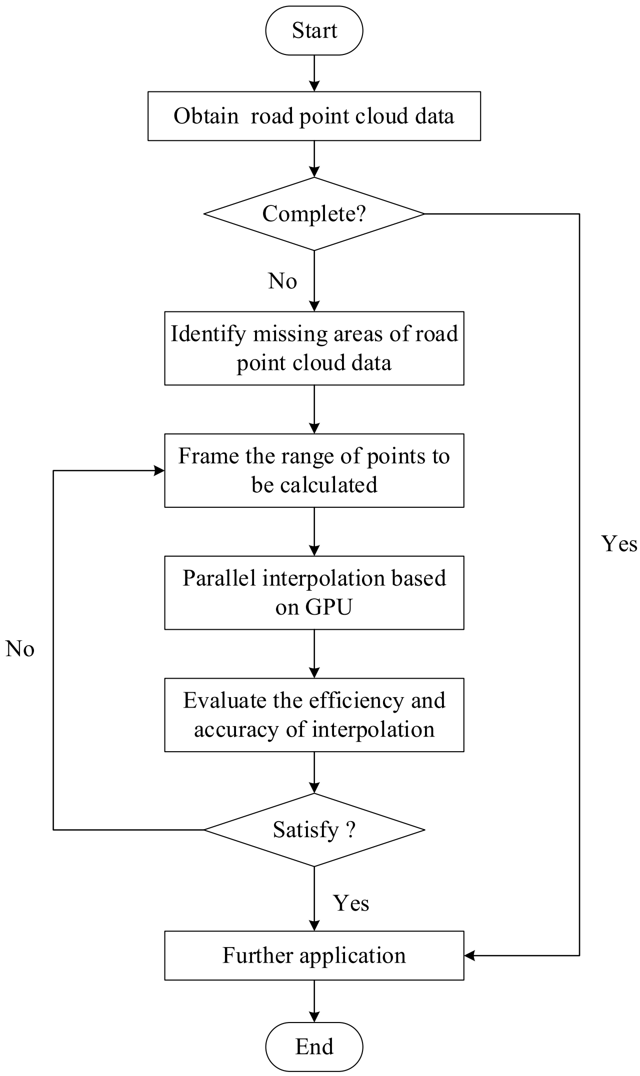


FIGURE 4. Process of incomplete road information imputation.

```
struct Pt {
  float x;
  float y;
  float z;
};
struct Pt myPts[N];
```

(a) AoS

```
struct Pt {
  float x[N];
  float y[N];
  float z[N];
};
struct Pt myPts;
```

(b) SoA

FIGURE 5. Two common data layouts: AoS and SoA [40].

shared memory, and then each thread in the thread block can access the coordinates that are stored in the shared memory at the same time. Each thread is responsible for calculating the distance and inverse weights of the data, and storing all the calculated weights and the results of the corresponding weights in two registers. Finally, according to the sum of the calculations of each thread, the value of each prediction point is obtained and transferred back to the global memory.

III. RESULT

A. EXPERIMENTAL ENVIRONMENT AND TESTING DATA

1) EXPERIMENTAL ENVIRONMENT

To evaluate the performance of the proposed method for incomplete road information imputation, two groups of experiments are conducted on a workstation computer. The specifications of the workstation are listed in Table 1.

TABLE 1. Specifications of the workstation computer employed for performing experimental tests.

Specifications	Details
CPU	Intel Xeon Gold 5118 CPU
CPU Frequency (GHz)	2.30
CPU RAM (GB)	128
CPU core	48
GPU	Quadro P6000
GPU memory (GB)	24
CUDA cores	3840
OS	Windows 10 Professor
Compiler	VS2015 Community
CUDA version	v9.0

2) TESTING DATA

The point cloud data obtained by LiDAR is a kind of three-dimensional data including position and elevation. We have designed two sets of simulation data sets with the same data attributes. In addition, we simulated the undulations of the real road surface, making the simulated data set closer to the actual field data set. To ensure that the method

TABLE 2. Summary information on testing data.

Dataset	Number of known data points on the road	Number of missing / interpolation points on the road	Remarks
1	14897392	98604	Uphill Scene
2	14897392	98604	Downhill Scene

proposed in this paper can be applied to the actual acquisition of field data sets.

To reflect real self-driving scenarios, as shown in Figure 6, we designed two groups of experiments. The experiment set up a two-way road with two lanes in each direction, and the total width of the road is 15 meters. The effective detection range of the designed LiDAR is 100 meters. We set up an obstacle vehicle. The vehicle model is simplified as a cuboid that is 5 meters long, 2 meters wide, and 1.6 meters tall. In addition, considering that the real road is sloped and rough, we create two scenes, i.e., the uphill and downhill. The gradient is 2%. That is, the height difference over a horizontal distance of 100 meters is 2 meters. We randomly generate approximately 15 million scattered points to simulate the point cloud data that are generated by the LiDAR over the testing road. A 6 millimeter oscillation is applied to the points on the same horizontal line to simulate the roughness of the real road surface.

FIGURE 6. Self-driving cars driving on sloped roads.

It should be noted that we specifically remove the points in the range of the obstacle vehicle's position to form a radar scanning blind spot area, ignore the heights of the points, and consider that the heights of the coordinate points at the position are unknown. That is, the road surface cloud data of the other vehicle position in the direction of the self-driving car are missing. More details of the two used datasets are presented in Table 2 and Figure 7.

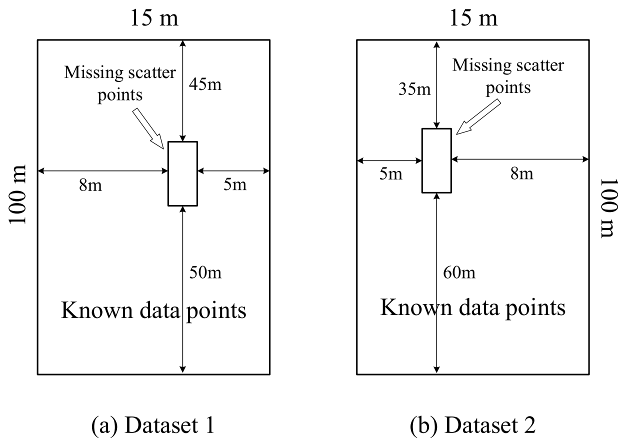


FIGURE 7. Schematic diagram of testing data.

B. EXPERIMENTAL RESULTS

There are two parameters that strongly affect the efficiency and accuracy of the imputation. The first is the extension range of the known data point cloud around the obstacle vehicle. A large extension range will lead to many data points, while a small range will result in too few data points. The second is the weight parameter of the IDW interpolation. The weight parameter of the IDW is related to the distance between the known data points and the unknown interpolation points. The larger the weight is, the greater the influence of the point near the missing point on the value of the missing point.

To determine a suitable extension range for the known data point cloud around the obstacle vehicle and the appropriate weight parameter of the IDW interpolation, we have carried out several groups of benchmark experiments, and the experimental results are shown in Table 3 to Table 8. For the sake of convenience, in those tables, the weight parameter is represented as P, and the configured range of known data points is represented as L.

When driving at high speed, self-driving cars need to identify and respond quickly to the road conditions in the direction of driving. Therefore, when interpolating an incomplete road point cloud using the IDW method, both the computational efficiency and accuracy must be guaranteed.

We compare the accuracy and efficiency of different IDW interpolation weights (P) and extension ranges (L), as shown in Figure 8 and Figure 9. It is found that with the increase of L, the number of known points and the interpolation accuracy increase, but the computational efficiency decreases accordingly. In addition, we find that when the weight of the IDW is 4, the calculation efficiency is relatively high. Its efficiency is higher than that when the weights are 3 and 5 but slightly lower than that when the weight parameter is 2.

Therefore, we choose to set the weight parameter to 4 for the IDW interpolation. Moreover, the influence of different extension ranges of the known data point cloud on the

TABLE 3. Experimental results of two groups of testing data when L=1.

Dataset	Running time (/ms)				RMSE(/mm)			
	P = 2	P = 3	P = 4	P = 5	P = 2	P = 3	P = 4	P = 5
1	138	148	140	144	57.79	56.32	54.87	53.43
2	155	151	150	157	57.67	56.18	54.72	53.32

TABLE 4. Experimental results of two groups of testing data when L=2.

Dataset	Running time (/ms)				RMSE(/mm)			
	P = 2	P = 3	P = 4	P = 5	P = 2	P = 3	P = 4	P = 5
1	222	259	217	227	51.03	46.13	42.03	38.79
2	225	227	222	231	50.92	46.02	41.92	38.68

TABLE 5. Experimental results of two groups of testing data when L=3.

Dataset	Running time (/ms)				RMSE(/mm)			
	P = 2	P = 3	P = 4	P = 5	P = 2	P = 3	P = 4	P = 5
1	334	347	342	370	41.95	33.38	27.38	23.55
2	334	324	332	341	41.79	33.20	27.20	23.43

TABLE 6. Experimental results of two groups of testing data when L=4.

Dataset	Running time (/ms)				RMSE(/mm)			
	P = 2	P = 3	P = 4	P = 5	P = 2	P = 3	P = 4	P = 5
1	520	540	466	472	32.20	20.29	13.87	10.46
2	478	486	478	483	32.13	20.11	13.88	10.41

TABLE 7. Experimental results of two groups of testing data when L=5.

Dataset	Running time (/ms)				RMSE(/mm)			
	P = 2	P = 3	P = 4	P = 5	P = 2	P = 3	P = 4	P = 5
1	654	668	658	682	24.39	10.67	5.08	3.54
2	660	658	657	664	24.10	10.81	5.00	3.63

TABLE 8. Experimental results of two groups of testing data when L=6.

Dataset	Running time (/ms)				RMSE(/mm)			
	P = 2	P = 3	P = 4	P = 5	P = 2	P = 3	P = 4	P = 5
1	847	849	844	897	17.59	6.57	4.01	4.01
2	849	842	842	844	17.61	7.76	4.30	4.11

interpolation efficiency and accuracy is analysed, and the results are shown in Figure 10.

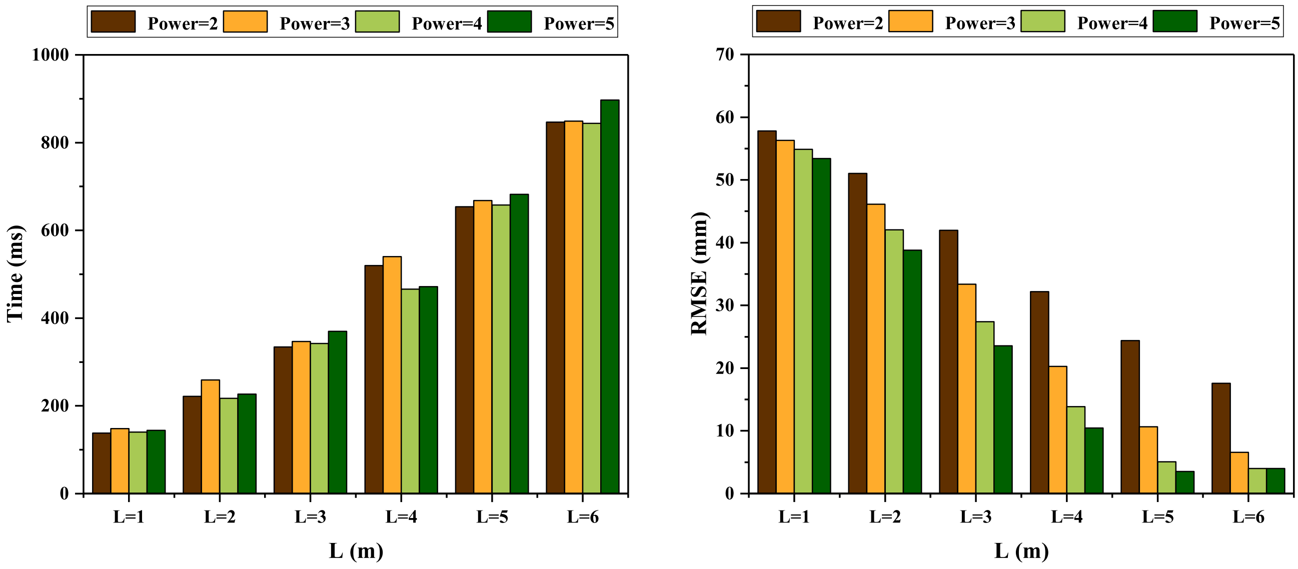
We find that the computational efficiency and accuracy have opposite trends. In Figure 10, it is found that the error can be controlled only when the extension range is 5 meters or 6 meters. When the range is less than 5 meters, there are too few points that are involved in the interpolation to reflect the road condition characteristics of the location. Thus, the error is large. The errors in the cases when the ranges are 5 meters and 6 meters are on the same magnitude, and the values of the errors are close to each other. However, when the range is 5 meters, the computational efficiency is much higher than that when the range is 6 meters.

Therefore, we suggest that when imputating the incomplete road point cloud data to enhance the safety of self-driving cars, the weight parameter of the IDW interpolation algorithm is recommended to be 4, and the extension range of the known data point cloud around the obstacle vehicle should be set as 5 meters.

IV. DISCUSSION

A. COMPARISON WITH OTHER METHODS

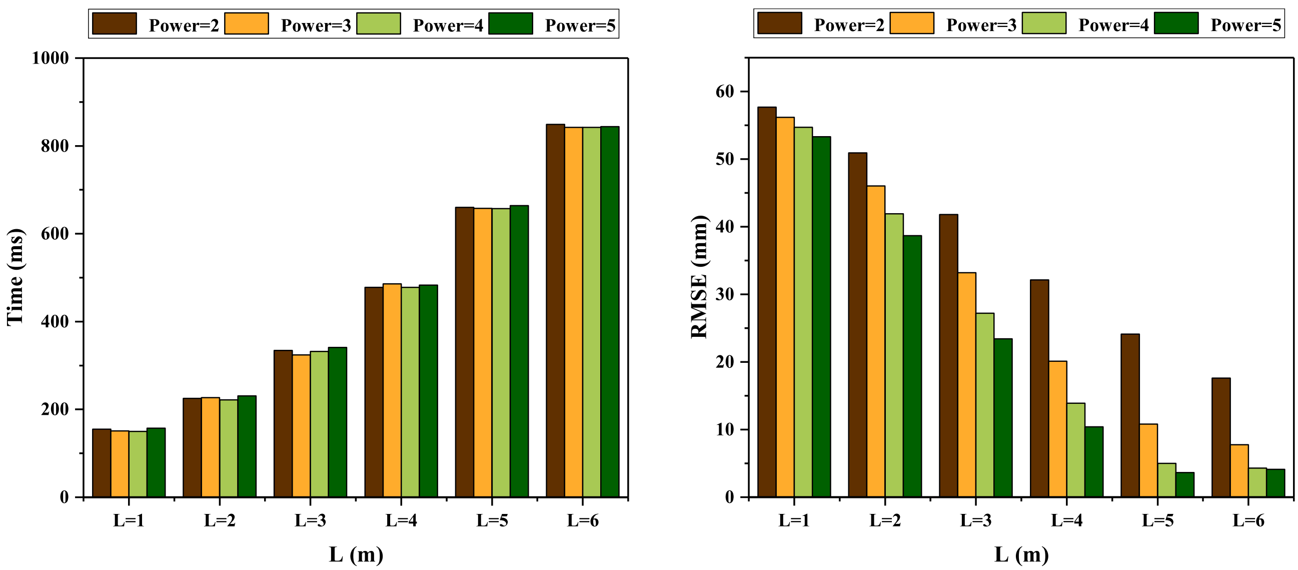
In the proposed method for incomplete road information imputation, we employ the GPU-accelerated parallel IDW algorithm to estimate the missing road point cloud. In this subsection, we will analyse the advantages of using the parallel IDW algorithm by comparing it with (1) machine learning algorithms such as the kNN algorithm and (2) other interpolation algorithms such as RBF and Moving Least Squares (MLS).



(a)

(b)

FIGURE 8. Results of Data Set 1. (a) Computational efficiency. (b) Computational accuracy.



(a)

(b)

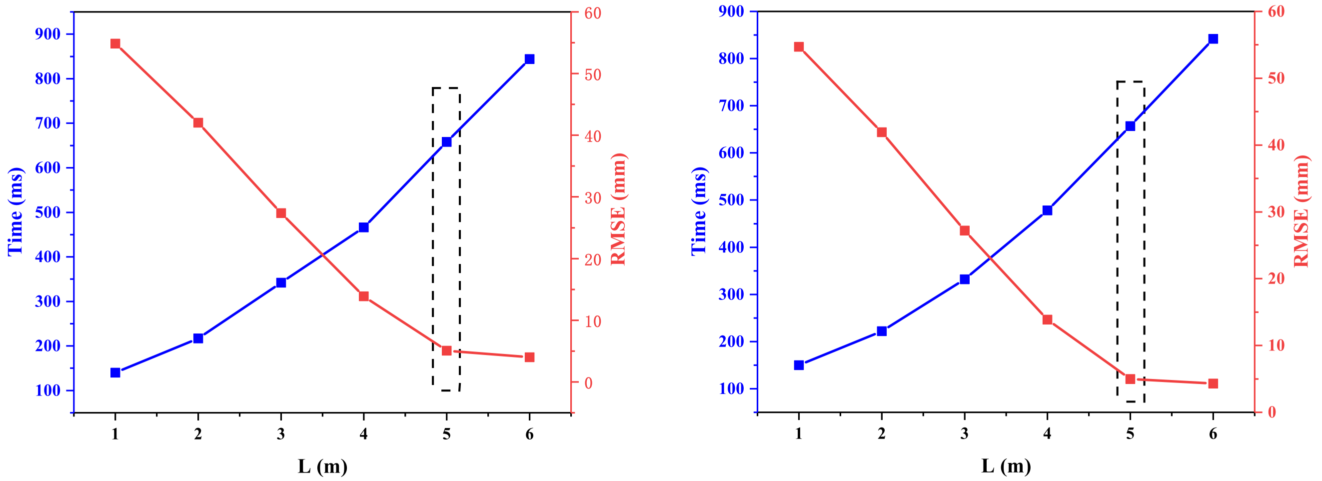
FIGURE 9. Results of Data Set 2. (a) Computational efficiency. (b) Computational accuracy.

1) COMPARISON WITH MACHINE LEARNING ALGORITHMS

Machine learning algorithms are widely used in various fields [43], [44]. By training known datasets, machine learning algorithms can estimate missing data. However, most machine learning techniques are computationally more expensive than interpolation algorithms [37], [45]. The commonly used machine learning algorithms for missing value estimations include the SVM, k-means, and deep neural networks. These algorithms are all executed iteratively. For large amounts of data, these methods cannot meet the requirements for computational efficiency. Among the machine learning algorithms, only lazy learning methods such as the kNN algorithm are quite competitive with respect to efficiency [46].

The kNN is one of the simplest machine learning algorithms. As a lazy learning algorithm, it does not need iterative calculations, and it only needs the corresponding weight to estimate the missing data. In this paper, we search k known points near each of the missing points as the computing data set. The Gauss weight is used to calculate the value of each missing point. The estimated results are listed in Table 9.

With the increase of k, the time that is required for the kNN estimation exponentially increases, as shown in Figure 11.



(a) Accuracy and efficiency of dataset 1 at P=4.

(b) Accuracy and efficiency of dataset 2 at P=4.

FIGURE 10. Accuracy and efficiency of the two datasets at P=4.

TABLE 9. Experimental results of estimating missing data using *k*NN.

Dataset	Running time (/s)				RMSE(/mm)			
	k=100	*k*=200	*k*=500	*k*=1000	*k*=100	*k*=200	*k*=500	*k*=1000
1	0.926	4.024	19.430	64.140	4.198	4.247	4.351	4.476
2	0.895	3.874	18.752	58.695	4.338	4.396	4.514	4.657

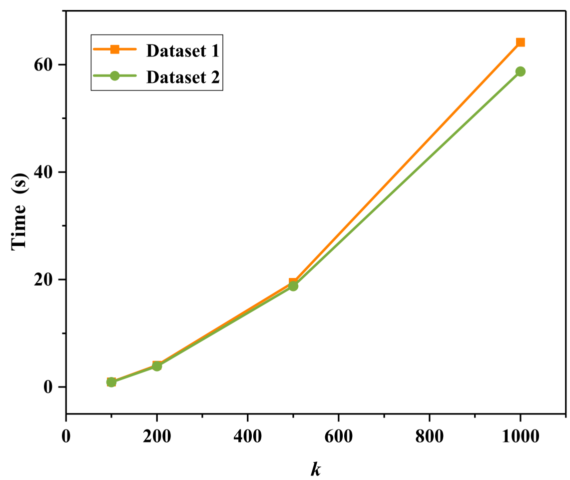


FIGURE 11. The computational efficiency of different *k* values (*k* represent the number of nearest neighbors).

Among them, most of the time is spent on searching the k nearest points around each of the missing points. Because the global trend of the data set that is used in the experiment is flat, the error when estimating the results is small, even if k is very small when using the kNN regression method. In fact, 100 or even 1000 known points near each missing point are not enough to reflect the real situation of the whole road surface. If intending to reflect the actual road conditions, k must be set to a larger number. However, this will become computationally much more expensive, as shown in Table 9.

Therefore, we suggest that in autonomous driving applications, the GPU-accelerated parallel IDW interpolation algorithm is more suitable for estimating the missing data of road points than the commonly used machine learning algorithms.

2) COMPARISON WITH OTHER INTERPOLATION ALGORITHMS

Other commonly used interpolation algorithms for estimating missing data include the MLS, RBF, adaptive IDW, and Kriging algorithms. Among them, the MLS needs to construct a matrix to solve when approximating the known data points [36]. If all the known point data are used to construct the matrix, the matrix will be too large to be stored. Therefore, it is necessary to search the k nearest points to the missing points using the kNN method to solve them. However, a similar problem will occur as in the kNN regression method. I.e., a large value of k will result in unsatisfactory computational efficiency.

RBF interpolation is one of the interpolation methods that can achieve satisfactory accuracy. The commonly used radial basis function is a Gauss function. When calculating it, it is also necessary to construct a matrix and solve it using the Gauss elimination method. The RBF interpolation method is obviously not applicable to a large number of data points that are produced by the LiDAR detector of the self-driving car. Similarly, the Kriging method needs to solve a system of linear equations when calculating the weights, which is not suitable for estimating a large number of missing points. The adaptive IDW (AIDW) method is an improved version of the IDW algorithm [42]. The essential idea behind the

AIDW is to adaptively calculate the inverse distance weight. In the adaptation process, k points around each missing point need to be searched, and k values must be set to a larger number to reflect the actual road conditions. However, this will obviously encounter the same problems as the kNN regression method.

Although more complex algorithms can produce more accurate results, they usually require higher computational costs [47]. In the application of autonomous driving, rapid environmental awareness and decision making are the most important technical problems. The parallel IDW interpolation based on GPUs is more efficient than other commonly used interpolation algorithms, and it also provides satisfactory interpolation accuracy.

B. LIMITATIONS OF THE USE OF POINT CLOUD DATA PRODUCED BY LIDAR

The main tasks of the environmental perception of self-driving cars include extracting road information, detecting obstacles, and calculating the positions of obstacles relative to vehicles. By scanning road information, LiDAR sensors can obtain high-precision road point cloud information map. However, LiDAR scanning is sensitive to weather conditions. In rainy, foggy, or snowy weather, the performance of LiDAR is not ideal. In addition, LiDAR cannot detect small obstacles, such as traffic signs that are 60 meters away. Because they occupy a lower scanning angle than the resolution of the LiDAR, the LiDAR cannot detect such obstacles.

C. LIMITATIONS OF THE EMPLOYED GPU IN THIS PAPER

IDW interpolation algorithm has good parallelism. In the actual situation, the CPU evenly distributes the calculation work to the GPU, and our work is mainly performed in the GPU. Therefore, our focus is mainly on GPU performance. In order to solve the shortcomings of the performance of a single GPU, the use of multiple GPUs has been proposed. Many scholars and enterprises have conducted research on multi-GPU interconnect technology [48]–[50]. The most representative of them are CrossFire [51] proposed by AMD and NVLink [52] and SLI [53] technology developed by NVIDIA.

NVIDIA has developed special processors for autonomous driving. NVIDIA DRIVE AGX Xavier [54] delivers 30 TOPS of performance while consuming only 30 watts of power. More importantly, NVIDIA recently introduced NVIDIA DRIVE AGX Orin [55], a highly advanced software-defined platform for autonomous vehicles and robots. The platform is powered by a new system-on-a-chip (SoC) called Orin, which consists of 17 billion transistors. The Orin SoC integrates NVIDIA's next-generation GPU architecture and Arm Hercules CPU cores, as well as new deep learning and computer vision accelerators that, in aggregate, delivers 200 TOPS of performance. Besides, NVIDIA DRIVE AGX Pegasus [54] achieves an unprecedented 320 TOPS of deep learning with an architecture built on two NVIDIA Xavier processors and two next-generation TensorCore GPUs. This energy-efficient, high-performance AI computer runs an array of deep neural networks simultaneously and is designed to safely handle highly automated and fully autonomous driving.

The computing capability of the single GPU (Quadro P6000 [56]) that we used in this paper is 12 TFLOPS. The computing resources on autonomous vehicles are more powerful than the workstation that we use. In practical situations, the calculation time required by the method used in this paper can meet the requirements of autonomous driving, and it can be applied to the real-time environmental perception of autonomous driving.

V. CONCLUSION

In this paper, we have proposed a fast method, which uses a GPU-based parallel IDW interpolation algorithm to impute the incomplete road point cloud datas obtained by IoT technology to enhance the safety of autonomous driving. Two groups of benchmarks have been conducted to evaluate the performance of the proposed method. We have found that: (1) the known point cloud data within 5 meters around the obstacle vehicle are sufficient to guarantee the imputation accuracy; (2) when the weight parameter of the IDW interpolation is 4, the efficiency and accuracy of the imputation can be optimally balanced; and (3) it takes approximately 0.6 seconds to imputate the incomplete dataset consisting of 15 million points, while the imputation error is approximately 5 millimeters. The proposed method is capable of efficiently and effectively imputating the incomplete road point cloud data that are induced by obstacle vehicles, and outperforms other interpolation algorithms and machine learning algorithms.

ACKNOWLEDGMENT

The authors would like to thank the editor and the anonymous reviewers for their constructive comments and suggestions, which improve the quality of this article.

LIST OF ABBREVIATIONS

AIDW	Adaptive Inverse Distance Weighted
AoS	Array of Structure
GPS	Global Positioning System
GPU	Graphics Processing Unit
IDW	Inverse Distance Weighted
IoT	Internet of Things
kNN	k-Nearest Neighbor
LiDAR	Light Detection And Ranging
MLS	Moving Least Squares
RBF	Radial Basis Function
RMSE	Root Mean Square Error
SoA	Structure of Array
SoC	System-on-a-chip
SVM	Support Vector Machine
TFLOPS	Trillion Floating-point Operations Per Second
TOPS	Trillion Operations Per Second

REFERENCES

[1] I. Farris, A. Orsino, L. Militano, A. Iera, and G. Araniti, "Federated IoT services leveraging 5G technologies at the edge," *Ad Hoc Netw.*, vol. 68, pp. 58–69, Jan. 2018.

[2] A. Orsino, G. Araniti, L. Militano, J. Alonso-Zarate, A. Molinaro, and A. Iera, "Energy efficient IoT data collection in smart cities exploiting D2D communications," *Sensors*, vol. 16, no. 6, p. 836, Jun. 2016.

[3] G. Mei, N. Xu, J. Qin, B. Wang, and P. Qi, "A survey of Internet of Things (IoT) for geo-hazards prevention: Applications, technologies, and challenges," *IEEE Internet Things J.*, to be published.

[4] A. Akbar, A. Khan, F. Carrez, and K. Moessner, "Predictive analytics for complex IoT data streams," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1571–1582, Oct. 2017.

[5] L. Shu, M. Mukherjee, M. Pecht, N. Crespi, and S. N. Han, "Challenges and research issues of data management in IoT for large-scale petrochemical plants," *IEEE Syst. J.*, vol. 12, no. 3, pp. 2509–2523, Sep. 2018.

[6] L. Catarinucci, D. De Donno, L. Mainetti, L. Palano, L. Patrono, M. L. Stefanizzi, and L. Tarricone, "An IoT-aware architecture for smart healthcare systems," *IEEE Internet Things J.*, vol. 2, no. 6, pp. 515–526, Dec. 2015.

[7] M. Hassanalieragh, A. Page, T. Soyata, G. Sharma, M. Aktas, G. Mateos, B. Kantarci, and S. Andreescu, "Health monitoring and management using Internet-of-Things (IoT) sensing with cloud-based processing: Opportunities and challenges," in *Proc. IEEE 12th Int. Conf. Services Comput.*, Jun. 2015, pp. 285–292.

[8] B. Xu, L. Da Xu, H. Cai, C. Xie, J. Hu, and F. Bu, "Ubiquitous data accessing method in IoT-based information system for emergency medical services," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1578–1586, May 2014.

[9] C. Du and S. Zhu, "Research on urban public safety emergency management early warning system based on technologies for the Internet of Things," *Procedia Eng.*, vol. 45, pp. 748–754, Jan. 2012.

[10] B. Kantarci and H. T. Mouftah, "Trustworthy sensing for public safety in cloud-centric Internet of Things," *IEEE Internet Things J.*, vol. 1, no. 4, pp. 360–368, Aug. 2014.

[11] T. U. Saeed, R. Nateghi, T. Hall, and B. S. Waldorf, "Statistical analysis of area-wide alcohol-related driving crashes: A spatial econometric approach," *Geograph. Anal.*, p. 24, 2019.

[12] S. A. Bagloee, M. Tavana, M. Asadi, and T. Oliver, "Autonomous vehicles: Challenges, opportunities, and future implications for transportation policies," *J. Mod. Transp.*, vol. 24, no. 4, pp. 284–303, Dec. 2016.

[13] C. Katrakazas, M. Quddus, W.-H. Chen, and L. Deka, "Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions," *Transp. Res. C, Emerg. Technol.*, vol. 60, pp. 416–442, Nov. 2015.

[14] I. Y. Noy, D. Shinar, and W. J. Horrey, "Automated driving: Safety blind spots," *Saf. Sci.*, vol. 102, pp. 68–78, Feb. 2018.

[15] L. M. Hulse, H. Xie, and E. R. Galea, "Perceptions of autonomous vehicles: Relationships with road users, risk, gender and age," *Saf. Sci.*, vol. 102, pp. 1–13, Feb. 2018.

[16] M. D. Simoni, K. M. Kockelman, K. M. Gurumurthy, and J. Bischoff, "Congestion pricing in a world of self-driving vehicles: An analysis of different strategies in alternative future scenarios," *Transp. Res. C, Emerg. Technol.*, vol. 98, pp. 167–185, Jan. 2019.

[17] J. Vleugel and F. Bal, "More space and improved living conditions in cities with autonomous vehicles," *Int. J. Des. Nature Ecodyn.*, vol. 12, no. 4, pp. 505–515, Jan. 2018.

[18] L. Bai and S. Zhijiang, "Autonomous parking: A unified motion planning framework based on simultaneous dynamic optimization," in *Proc. 34th Chin. Control Conf. (CCC)*, Jul. 2015, pp. 5913–5918.

[19] G. De La Torre, P. Rad, and K. K. R. Choo, "Driverless vehicle security: Challenges and future research opportunities," *Future Gener. Comput. Syst.*, to be published.

[20] A. Nunes, B. Reimer, and J. Coughlin, "People must retain control of autonomous vehicles," *Nature*, vol. 556, no. 7700, pp. 169–171, Apr. 2018.

[21] J. Weyer, R. D. Fink, and F. Adelt, "Human–machine cooperation in smart cars. An empirical investigation of the loss-of-control thesis," *Safety Sci.*, vol. 72, pp. 199–208, Feb. 2015.

[22] Y. Cai, D. Li, X. Zhou, and X. Mou, "Robust drivable road region detection for fixed-route autonomous vehicles using map-fusion images," *Sensors*, vol. 18, no. 12, p. 4158, Nov. 2018.

[23] H. G. Seif and X. Hu, "Autonomous driving in the iCity—HD maps as a key challenge of the automotive industry," *Engineering*, vol. 2, no. 2, pp. 159–162, Jun. 2016.

[24] D. Dong, X. Li, and X. Sun, "A vision-based method for improving the safety of self-driving," in *Proc. 12th Int. Conf. Rel., Maintainability, Saf. (ICRMS)*, Oct. 2018, pp. 167–171.

[25] P. Drews, G. Williams, B. Goldfain, E. A. Theodorou, and J. M. Rehg, "Vision-based high-speed driving with a deep dynamic observer," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1564–1571, Apr. 2019.

[26] J. Jung and S.-H. Bae, "Real-time road lane detection in urban areas using LiDAR data," *Electronics*, vol. 7, no. 11, p. 276, Oct. 2018.

[27] G. Reina, A. Milella, R. Rouveure, M. Nielsen, R. Worst, and M. R. Blas, "Ambient awareness for agricultural robotic vehicles," *Biosyst. Eng.*, vol. 146, pp. 114–132, Jun. 2016.

[28] M. E. El Najjar and P. Bonnifait, "A road-matching method for precise vehicle localization using belief theory and Kalman filtering," *Auton. Robots*, vol. 19, no. 2, pp. 173–191, Sep. 2005.

[29] Z. J. Chong, B. Qin, T. Bandyopadhyay, M. H. Ang, E. Frazzoli, and D. Rus, "Synthetic 2D LiDAR for precise vehicle localization in 3D urban environment," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2013, pp. 1554–1559.

[30] A. Harrison and P. Newman, "High quality 3D laser ranging under general vehicle motion," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2008, pp. 7–12.

[31] C. Wang, Q. Sun, Z. Li, H. Zhang, and K. Ruan, "Cognitive competence improvement for autonomous vehicles: A lane change identification model for distant preceding vehicles," *IEEE Access*, vol. 7, pp. 83229–83242, 2019.

[32] T. Ni, W. Li, H. Zhang, H. Yang, and Z. Kong, "Pose prediction of autonomous full tracked vehicle based on 3D sensor," *Sensors*, vol. 19, no. 23, p. 5120, Nov. 2019.

[33] S. Wang, S. Kodagoda, L. Shi, and X. Dai, "Two-stage road terrain identification approach for land vehicles using feature-based and Markov random field algorithm," *IEEE Intell. Syst.*, vol. 33, no. 1, pp. 29–39, Jan. 2018.

[34] G. Reina, A. Milella, and R. Worst, "LiDAR and stereo combination for traversability assessment of off-road robotic vehicles," *Robotica*, vol. 34, no. 12, pp. 2823–2841, Dec. 2016.

[35] M. Holder, P. Rosenberger, H. Winner, V. P. Makkapati, M. Maier, H. Schreiber, Z. Magosi, T. D'Hondt, Z. Slavik, O. Bringmann, and W. Rosenstiel, "Measurements revealing challenges in radar sensor modeling for virtual validation of autonomous driving," in *Proc. IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 2616–2622.

[36] Z. Ding, G. Mei, S. Cuomo, H. Tian, and N. Xu, "Accelerating multi-dimensional interpolation using moving least-squares on the GPU," *Concurrency Comput., Pract. Exper.*, vol. 30, no. 24, p. e4904, Dec. 2018.

[37] Z. Ding, G. Mei, S. Cuomo, Y. Li, and N. Xu, "Comparison of estimating missing values in IoT time series data using different interpolation algorithms," *Int. J. Parallel Program.*, pp. 1–15, Aug. 2018.

[38] Z. Ding, G. Mei, S. Cuomo, N. Xu, and H. Tian, "Performance evaluation of GPU-accelerated spatial interpolation using radial basis functions for building explicit surfaces," *Int. J. Parallel Programm.*, vol. 46, no. 5, pp. 963–991, Oct. 2018.

[39] T. J. Klatko, T. U. Saeed, M. Volovski, S. Labi, J. D. Fricker, and K. C. Sinha, "Addressing the local-road VMT estimation problem using spatial interpolation techniques," *J. Transp. Eng., A, Syst.*, vol. 143, no. 8, Aug. 2017, Art. no. 04017038.

[40] G. Mei and H. Tian, "Impact of data layouts on the efficiency of GPU-accelerated IDW interpolation," *SpringerPlus*, vol. 5, no. 1, pp. 1–18, 2016.

[41] G. Mei, "Evaluating the power of GPU acceleration for IDW interpolation algorithm," *Sci. World J.*, vol. 2014, pp. 1–8, Feb. 2014.

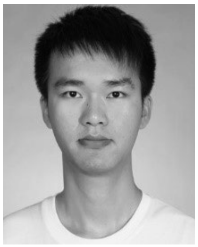
[42] G. Mei, N. Xu, and L. Xu, "Improving GPU-accelerated adaptive IDW interpolation algorithm using fast kNN search," *SpringerPlus*, vol. 5, no. 1, pp. 1–22, 2016.

[43] G. R. González and N. Arunkumar, "Introduction to the special section on hybrid artificial intelligence applications," *Comput. Elect. Eng.*, vol. 73, pp. 374–375, Jan. 2019.

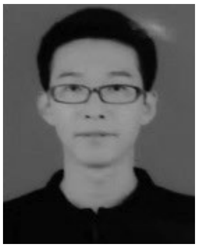
[44] L. Santamaria-Granados, M. Munoz-Organero, G. Ramirez-Gonzalez, E. Abdulhay, and N. Arunkumar, "Using deep convolutional neural network for emotion detection on a physiological signals dataset (AMIGOS)," *IEEE Access*, vol. 7, pp. 57–67, 2019.

[45] C.-F. Tsai, M.-L. Li, and W.-C. Lin, "A class center based approach for missing value imputation," *Knowl.-Based Syst.*, vol. 151, pp. 124–135, Jul. 2018.

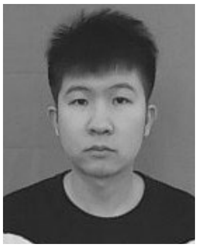
[46] D. J. Stekhoven and P. Buhlmann, "MissForest-non-parametric missing value imputation for mixed-type data," *Bioinformatics*, vol. 28, no. 1, pp. 112–118, Jan. 2012.
[47] I. B. Aydilek and A. Arslan, "A hybrid method for imputation of missing values using optimized fuzzy c-means with support vector regression and a genetic algorithm," *Inf. Sci.*, vol. 233, pp. 25–35, Jun. 2013.
[48] D. Datta, D. Appelhans, C. Evangelinos, and K. Jordan, "An asynchronous two-level checkpointing method to solve adjoint problems on hierarchical memory spaces," *Comput. Sci. Eng.*, vol. 20, no. 4, pp. 39–55, Jul. 2018.
[49] J. Choquette, O. Giroux, and D. Foley, "Volta: Performance and programmability," *IEEE Micro*, vol. 38, no. 2, pp. 42–52, Mar. 2018.
[50] D. Appelhans, G. Auerbach, D. Averill, R. Black, A. Brown, D. Buono, R. Cash, D. Chen, M. Deindl, D. Duffy, and G. Eastman, "Functionality and performance of NVLink with IBM POWER9 processors," *IBM J. Res. Develop.*, vol. 62, nos. 4–5, p. 10, 2018.
[51] *AMD Crossfire Technology*. Accessed: Dec. 15, 2019. [Online]. Available: https://www.amd.com/en/technologies/crossfire
[52] *NVLink and NVSwitch*. Accessed: Dec. 15, 2019. [Online]. Available: https://www.nvidia.com/en-us/data-center/nvlink/
[53] *NVIDIA GameWorks Documentation*. Accessed: Dec. 15, 2019. [Online]. Available: https://docs.nvidia.com/gameworks/content/technologies/desktop/sli.htm
[54] *NVIDIA Self-Driving Cars*. Accessed: Dec. 15, 2019. [Online]. Available: https://www.nvidia.com/en-us/self-driving-cars/drive-platform/hardware/
[55] *NVIDIA Software Platform for Autonomous Machines*. Accessed: Dec. 15, 2019. [Online]. Available: https://nvidianews.nvidia.com/news/nvidia-introduces-drive-agx-orin-advanced-software-defined-platform-for-autonomous-machines
[56] *NVIDIA Quadro Desktop GPU*. Accessed: Dec. 15, 2019. [Online]. Available: https://www.nvidia.com/en-us/design-visualization/quadro-desktop-gpus/



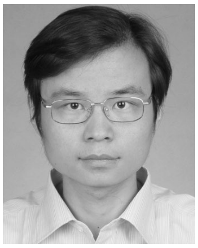
KAIFENG GAO is currently pursuing the Ph.D. degree with the China University of Geosciences (Beijing). His research interests are in the areas of the Internet of Things, machine learning, and computational modeling.



BOWEN WANG is currently pursuing the Ph.D. degree with the China University of Geosciences (Beijing). His research interests are in the areas of 3D geological modeling and numerical simulation, including FEM analysis, computational geometry, and Delaunay triangulation.



LEI XIAO is currently pursuing the Ph.D. degree with the China University of Geosciences (Beijing). His research field interests are in the areas of numerical simulations and computational modeling, complex science, and GPU computing.



GANG MEI (Member, IEEE) received the bachelor's and master degrees from the China University of Geosciences (Beijing), and the Ph.D. degree from the University of Freiburg, Germany, in 2014. He is currently an Associate Professor in numerical modeling and simulation with the China University of Geosciences (Beijing). His main research interests are in the areas of numerical simulation and computational modeling, GPU computing, data mining, and network science and applications. He has published more than 50 research articles in journals and academic conferences. He has been serving as an Associate Editor for the journal IEEE Access, since 2018.