

Article

A Three-Dimensional Visualization Framework for Underground Geohazard Recognition on Urban Road-Facing GPR Data

Zhenwu Wang ^{1,*} , Benting Wan ² and Mengjie Han ³ 

¹ Department of Computer Science and Technology, China University of Mining and Technology, Beijing 100083, China

² School of Software and IoT Engineering, Jiangxi University of Finance & Economics, Nanchang 330013, China; wanbenting@jxufe.edu.cn

³ School of Technology and Business Studies, Dalarna University, 79188 Falun, Sweden; mea@du.se

* Correspondence: wzw@cumtb.edu.cn

Received: 29 September 2020; Accepted: 9 November 2020; Published: 11 November 2020



Abstract: The identification of underground geohazards is always a difficult issue in the field of underground public safety. This study proposes an interactive visualization framework for underground geohazard recognition on urban roads, which constructs a whole recognition workflow by incorporating data collection, preprocessing, modeling, rendering and analyzing. In this framework, two proposed sampling point selection methods have been adopted to enhance the interpolated accuracy for the Kriging algorithm based on ground penetrating radar (GPR) technology. An improved Kriging algorithm was put forward, which applies a particle swarm optimization (PSO) algorithm to optimize the Kriging parameters and adopts in parallel the Compute Unified Device Architecture (CUDA) to run the PSO algorithm on the GPU side in order to raise the interpolated efficiency. Furthermore, a layer-constrained triangulated irregular network algorithm was proposed to construct the 3D geohazard bodies and the space geometry method was used to compute their volume information. The study also presents an implementation system to demonstrate the application of the framework and its related algorithms. This system makes a significant contribution to the demonstration and understanding of underground geohazard recognition in a three-dimensional environment.

Keywords: underground geohazards; interactive visualization framework; GPR; Kriging algorithm

1. Introduction

In the last decade, the visualization platforms of above-ground geographical information have been constructed in various fields [1–5]. Owing to the concealment of the underground space, detecting underground geohazards on urban roads is a difficult problem across the world. The difficulty lies in establishing a thorough workflow and integrated platform of data acquisition, data preprocessing, data interpretation, three-dimensional (3D) modeling and visualization of the interpreted results. Every year it is reported that extensive and expensive property damages were caused by road collapse mainly derived from various underground geohazards [6]. The collapse accidents in urban areas can not only threaten human lives and properties but can also lead to damage of adjacent infrastructures, such as communication or drainage facilities, in the subsurface space surrounding the collapse area [7]. Although ground penetrating radar (GPR) technology enables geologists to explore the evolution mechanism and adopt measures to avoid underground geohazards on urban roads, enhancing the quality of preprocessed GPR data, where the generated data can affect the accuracy of geohazard

recognition directly, is still a considerable problem. In addition, 3D modeling and visualization of underground geohazards can provide convenience for geohazards presentation and analysis. With the idea of addressing some of these limitations, some research articles [8–24] have discussed different ways to visualize the information from underground space. These have made important contributions to the architectural design of underground information systems and the related representation methods of underground structures. However, they can only provide a model demonstration of the operation process of underground infrastructures from the existing design and construction drawings [8,14,15,19–22]. In reality, these demonstration plans of underground infrastructures (for example, pipelines) may be quite different from the actual conditions in the ground owing to the development and change of urban roads. Some works [9,10,17,18,23] used the GPR data to visualize two-dimensional space, but their systems cannot construct intuitive three-dimensional (3D) scenes nor can they assess the countermeasures accurately. Recently, Mauricio [24] proposed a processing workflow of GPR data to create 3D models of subsurface tubular structures, which adopted image reconstruction and enhancement technologies. Other research [11–13,16] adopted three-dimensional GPR to construct 3D scenes that used commercial software.

Inspired from previous research and the functionalities of commercial software, we propose a comprehensive three-dimensional framework to organize the workflow of underground geohazard detection. Based on our research team's self-developed GPR equipment, this study presents an interactive 3D visualization framework for underground geohazard recognition on urban roads called 3DVF4UDR, which complements the existing GPR instrument and visualization platforms profoundly. The developed framework can provide practical implementation details to geoinformation engineers. The main contributions of this paper are that:

- (1) We presented a framework that integrates the detection processes into one workflow, including data acquisition, preprocessing, modeling, visualization and interactive geohazard recognition and analysis. The data acquisition is based on the self-developed GPR equipment. Data formats and the loosely coupled formats make the framework load data in a flexible way. The details are described in Section 3.1.
- (2) In this framework, a series of proposed novel algorithms provided the theoretical support for the recognition processes, including the data selection methods of the Kriging algorithm, the improved GPU-PSO-Kriging algorithm and the layer-constrained triangulated irregular network (TIN) algorithm. These can not only make the GPR data preprocessing faster and more accurate, but can also construct geohazard bodies of arbitrary shapes either in part or as a whole. A detailed description of this framework is formulated in Section 3.2, Section 3.3, Section 3.4 and the visualization results are described in Section 4.
- (3) In this work, we did not only provide a theoretical analysis and experimental verification for the framework with related algorithms, but also developed a platform system to test the effectiveness of the proposed framework and its algorithms in a practical dimension.

The rest of this work is organized as follows. Section 2 presents an overview of the existing work on GPR data acquisition, data preprocessing, and 3D modeling and visualization methods. Comprehensive descriptions of 3DVF4UDR are discussed in Section 3, where the proposed algorithms in different workflow steps are discussed in detail while Section 4 demonstrates the system implementation of 3DVF4UDR, which consolidates the proposed novel algorithms and shows the applied effects of these algorithms. Section 5 offers a conclusion to the research and discusses the potential for future work.

2. Existing Works

In this section, we discuss research works that are related to the adopted technologies used in 3DVF4UDR. The 3DVF4UDR acquires its data through GPR instruments. Section 2.1 introduces the mainstream instruments and discusses the advantages of using our self-developed equipment. The raw GPR data should be preprocessed for further modeling. Section 2.2 discusses the Kriging algorithm

and outlines our proposed methods. For 3D geological modeling, the mainstream geological models and our proposed algorithm are discussed in Section 2.3.

2.1. GPR Data Acquisition

Due to the high-speed construction of urban roads during the last few decades, road maintenance has become an extremely important issue. Road collapse is one of the main problems and can demonstrate many different characteristics, such as concealment, paroxysm, recurrence and group-occurring [6]. Collapses are mainly caused by various geohazards under the road subgrade, which can be hard to detect. Effective road maintenance needs urgent and cost-effective inspection methods to reduce road deterioration. GPR is the main technology for road subsurface inspection [9], because it can acquire general survey data in a relatively short period of time. Different GPR devices have been designed by different companies and research institutions, for example, SIR series GPR (Geophysical Survey Systems Inc., GSSI), Pulse EKKO series GPR (Canada Sensor & Software Inc., SSI), Explore GPR (Italy INGEGNERIA DEI SISTEMI (IDS) Company), LTD GPR (China Research Institute of Radio Wave Propagation, CRIRP), and GR-Radar series GPR (our research team).

As depicted in Figure 1, our GPR sends high-frequency electromagnetic pulse toward underground. A two-way travel time, 50 nanoseconds for our GPR, is the time between leaving the sender and returning to the receiver for the pulse. When the pulse meets a dielectric interface with different relative permittivity during the propagation, a part of the pulse will be reflected to the surface and the reflected wave can also be received by the receiver. During the 50 nanoseconds, the reflected waves will be received 512 (or 1024) times with equal time interval and generate 512 (or 1024) sample points. After all 512 (or 1024) sample points having been collected at a horizontal position, the GPR collects new samples at next horizontal position until all positions are reached within a pre-defined area. Our GPR has eight channels [25] and the detection speed can reach 60km per hour, which are comparable to other GPR devices. Data acquired and format will be discussed in Section 3.1.3.

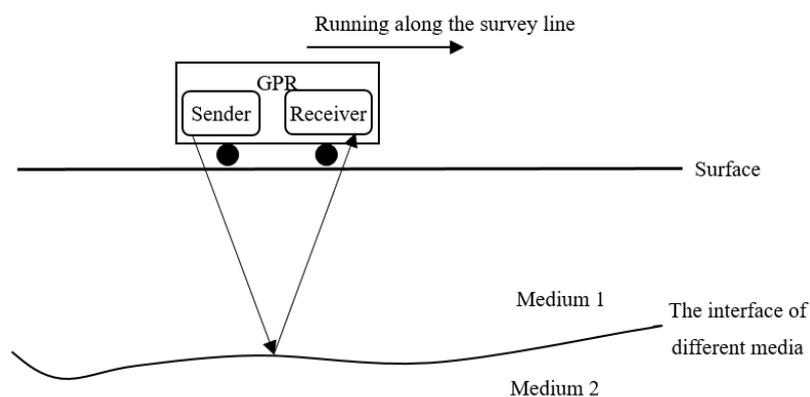


Figure 1. Detection principle of GPR.

2.2. GPR Data Preprocessing

Generally speaking, GPR just generates the data of survey line that should be preprocessed in order to obtain more comprehensive data that covers the whole detected area. Of all of the preprocessed methods used to reveal the theory of spatial structure correlation and randomness [26], the Kriging algorithm is one of the most widely used technologies. The current research on this algorithm focuses on the combination of different types of Kriging methods [27–30], or focuses on the hybridization between Kriging and other algorithms, such as bionic optimization algorithm [31–33], the support vector machine method [34], the least-square method [35], and the Gradient information [36]. To our knowledge, only a few articles have discussed how to select data points for interpolation, which is the prerequisite affecting whether the Kriging interpolation results are accurate or not. We have proposed two novel sample points selection algorithms (named D3DGC and DRD) in a previous work [37] that

is to be integrated in this study. In addition, we also propose a GPU-PSO-Kriging algorithm that uses a Compute Unified Device Architecture (CUDA) to run a particle swarm optimization (PSO) algorithm in parallel and a PSO that optimizes the parameters in the Kriging algorithm in order to interpolate data more accurately. All of these activities are discussed in Section 3.2.

2.3. 3D Geological Modeling

Three-dimensional geological modeling is an abstract process involving the reconstruction and reproduction of geological bodies through 3D visualization technologies [38]. The complexity of geological spatial relationships increases the complexity of the data structure, topological relationship and corresponding algorithms, which makes geological modeling very difficult [39]. There are many 3D geological models [40,41], such as triangulated irregular network (TIN), triangular prism (TP), analogical triangular prism (ATP), generalized triangular prism (GTP), analogical right triangular prism (ARTP), tetrahedral network (TEN), and so on. Compared with other models, TIN cannot only change the size and number of triangular patches according to the complexity of the geological surface, but can also eliminate the data redundancy in the visualization step and maintain a high fitting accuracy [37]. Current research on TIN's construction [42–48] assumes that the point set of the discrete data is static and rarely discusses how to manage dynamic data points in different levels, which means that all the data points are handled together rather than creating connections layer by layer. A layer constrained TIN (LC-TIN) algorithm was proposed in a previous work [49] to construct 3D bodies from layer to layer. The details of the LC-TIN are discussed in Section 3.3.

3. The Principle of 3DVF4UDR

In this section, we begin by giving an overall description of 3DVF4UDR, including the architecture, the corresponding workflow and the definition of file formats. The proposed algorithms in 3DVF4UDR are then discussed in-depth in the following workflow steps.

3.1. Overall Description of 3DVF4UDR

The layered architecture of 3DVF4UDR makes it flexible so that it can configure the workflow steps. The self-defined file formats ensure 3DVF4UDR can conveniently save or load the interpreted results.

3.1.1. The Architecture

As described in Figure 2, the architecture of 3DVF4UDR adopts the specification of Model-View-Controller (MVC) including resources layer, services access layer, business logic layer and presentation layer. The layer division can encapsulate different program functions into different containers that give 3DVF4UDR high flexibility and maintainability.

The presentation layer encapsulates the framework's interactive interface to users. The GPR's original and intermediate or final result data are displayed in 2D/3D views and the framework functions are organized as different type windows. The business logic layer is responsible for system logic dispatch. For example, the algorithm scheduling component is in charge of dispatching a series of algorithms, including *GPU-PSO-Kriging*, *LC-TIN*, *D3DGC*, *DRD*, and so on. The rendering engine is implemented by OpenGL that can execute the 2D/3D rendering and CUDA computing. The self-defined tools component has an *ArcBall* sub-component that can control the moving direction and zoom of 3D objects, a *MathUtils* sub-component to realize common mathematical functions, the self-developed *RichTree* and *RichList* control sub-components. The data, model, process and rendering management components are derived and dispatched in the management container. In the services access layer, there is an input/output interface of different original, intermediate, or final data. Various basic and complex data types are defined in the data type container. Finally, the raw, processed and management data is stored in a database or file system in resources layer.

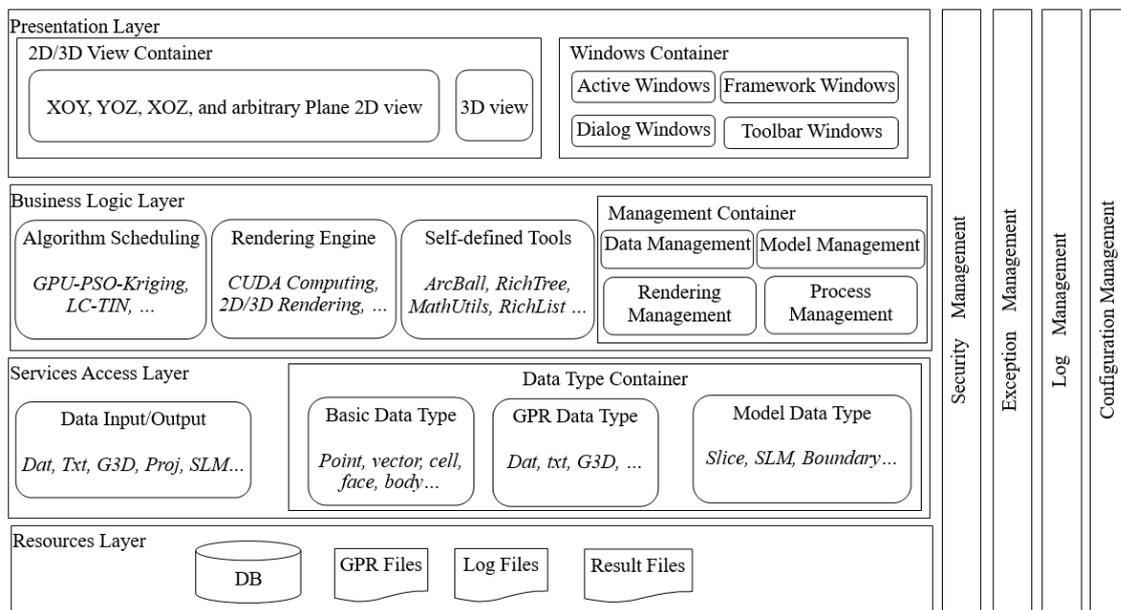


Figure 2. The architecture of 3DVF4UDR.

3.1.2. Workflow Description

The 3DVF4UDR integrates the whole process of underground geohazard recognition. The workflow can be described in Figure 3, which contains all of the operations from data acquisition to result generation, with the previous process acting as the prerequisite for the next process. The data in each step can be visualized by the rendering engine, including raw GPR data (saved as *.txt* and *.dat* files), interpolated data (*.G3D* files), slice data (*.slc* and *.slm* files), and the final result images. Different file formats can analyze the original, intermediate and final results in a more convenient way. They organize data in the services access layer and ensure that the system can start from any step in the operation.

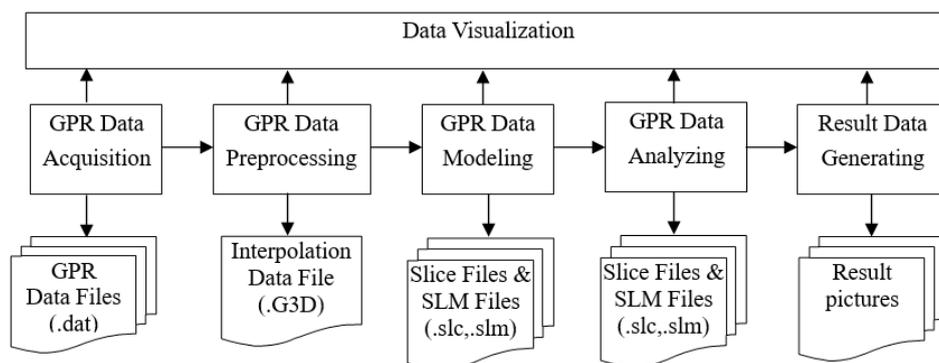


Figure 3. The workflow of 3DVF4UDR.

3.1.3. File Format Descriptions

In the 3DVF4UDR, some self-defined file formats are used to describe the GPR source and intermediate data, including *.txt*, *.dat*, *.G3D*, *.slc*, and *.slm*. As described in Figure 4a, a survey line index file (*.txt* file) is in charge of organizing a survey line data file (*.dat* file) for enhancing the transfer efficiency. The *.dat* file is saved as binary data that only contains the values of amplitudes from dielectric interface collected by an electromagnetic wave. It is defined by its start (X_s, Y_s) and end (X_e, Y_e) coordinates in Figure 4a. In addition, the uniform information of all of the survey line data is defined in *.txt* files, such as the number of horizontal positions and sample points, electromagnetic wave speed and two-way travel time. As described in Figure 4d, survey line data files can be interpolated into a *.G3D* file

that contains all of the information covered by the detection area. The file head of *.G3D* includes the minimum and maximum value information in *X*, *Y*, *Z* directions and the number of interpolated points in the three directions that are defined as *IPNX*, *IPNY* and *IPNZ*. The body of the *.G3D* file stores all of the data that are interpolated along the survey lines (Figure 1) according to the obtaining sequence of the sample points. The geohazard information is saved in a slice file (*.slc* file). This data is selected by broken lines that are mapped to slice data. The slice data is then extracted from the *.G3D* data. As shown in Figure 4c, the proposed framework supports to identify different geohazard bodies, which are defined by the different headings “Body Type No.”, “Body No.” and “Body attributes”. “Body attributes” refers to the body types of different geohazards, for example, underground cavities, water, or cracks. “Body No.” is the geohazard body serial number. Both “Body No.” and “Body Type No.” can uniquely determine a certain type of geohazard body. Each body has its own attribute information, such as body color, transparency, font, and so on. All of the slice files are managed by the *.slm* file that is defined in Figure 4b. “Slice direction” indicates that the 3DVF4UDR can divide *.G3D* data into various slices according to different coordinate planes, such as *XOY*, *YOZ*, and *XOZ* planes. Other statistical information (number of slices and 3D geohazard bodies’ amount) is also described in the slice management file.

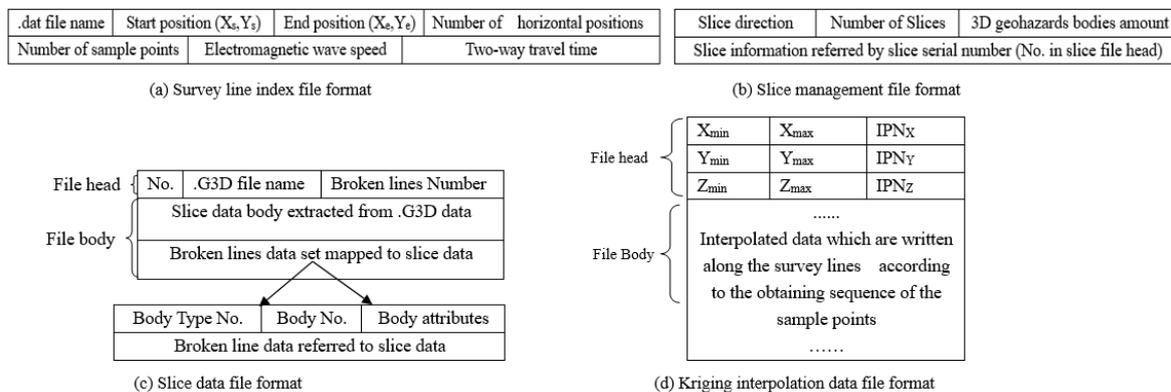


Figure 4. The file formats in 3DVF4UDR.

According to the aforementioned discussion, except the raw and interpolated data file (*.dat* and *.G3D* files), the intermediate data files (for example, the *.slc* and *.slm* files) can make the 3DVF4UDR load generate geohazard bodies at any time, which can enhance operational efficiency.

The following sections will discuss some of the algorithms related to the workflow in more details. For the data preprocessing step, 3DVF4UDR adopts *D3DGC* and *DRD* algorithms to choose the sampling data for the Kriging algorithm more reasonably and uses a particle swarm optimization (PSO) algorithm to optimize the Kriging parameters in a GPU parallel mode (named GPU-PSO-Kriging algorithm). In the process of data modeling, the LC-TIN algorithm was designed to model the geohazard bodies from layer to layer as arbitrary shapes. A space geometric algorithm was applied to compute the volume information of the geohazard bodies in this step of the data analysis.

3.2. Data Preprocessing

Generally, survey line data generated by a GPR instrument should be preprocessed to generate adequate data. The Kriging interpolation algorithm is widely applied to perform this task. A key problem is then how to select sampling data for the Kriging method. Our study adopted a GPU-PSO-Kriging interpolation algorithm to handle the survey line data. Two sampling points selection algorithms were used in the framework to enhance interpolation accuracy, namely a dynamic three-dimensional globe cover algorithm (named *D3DGC*) and a double reverse distance algorithm (*DRD*) [37].

3.2.1. Sampling Point Selection Algorithms

The rational selection of sampling data is a prerequisite for ensuring the accuracy of the Kriging interpolation results. Liu et al. [50] suggested that some factors could affect interpolation results, including the distribution characteristics of sampled data, the nugget effect, the range and type of variation model, as well as the anisotropy and searching range of space configuration. Deng et al. [51] further pointed out that the sampled data that was close to estimation point had a great influence on the estimation structure and that a spherical model and exponential model should be chosen in most cases. However, they only offered a general rule of sampled data selection and did not discuss how to select sampling data deeply. Based on their findings on minimum variance criterion, the work [26] used multi-distance measures to describe space structures, including Manhattan distance, Euclidean distance and Chebyshev distance measures. However, their research only extended the distance measure and did not discuss the sampled data selection problem.

According to the theory of geological data correlation, the points that are close to the interpolated point should be selected as the sampling points. The D3DGC algorithm generates sampling points by moving a sphere area that took the interpolated point as the center and had a radius of dynamic growth until it satisfied the quantity requirement of sampling points. When the values of survey line data are similar and the data volume is very large, the interpolation effects may not be the best approach when only considering on space correlation theory. By introducing randomness, the proposed DRD method allocates points to different survey line data and further arranged point distribution on each survey line according to proposed correlation and randomness rules.

It has been elaborated that D3DGC algorithm can figure out the best number of sample points, e.g., 210, given a specific dataset [37]. It reduces the error rate from 8% to 1.3% compared with traditional method that only uses horizontal neighbours for interpolation. When the number is taken between 180 and 270 rather than 210, the error rate will be only increasing up to 10% for D3DGC. On the other hand, traditional methods can only reach 10% or lower for a small fraction of selection numbers and can get as much as 31%. The reason is that D3DGC considers the correlations between the sample points and it searches the sample space in a spherical area by gradually increasing the radius. Thus, the complexity of D3DGC is $O(m * n * p * u)$, where m, n, p, u are the number of sample points, total points, survey lines and horizontal positions, respectively. Another option of sample points selection algorithm is DRD that also gives a lower error rate. DRD has a complexity of $O(n * p)$, which is suitable for sampling from large data.

3.2.2. GPU-PSO-Kriging Algorithm

The Kriging interpolation algorithm is the best unbiased linear space prediction method [52], which is based on the variation distribution of space attributes on space positions. Suppose that there are N sampling points pt_i , $1 \leq i \leq N$ and Equation (1) defines the ordinary Kriging prediction:

$$Z^*(pt_0) = \sum_{i=1}^N \lambda_i Z(pt_i), \quad (1)$$

where $Z^*(pt_0)$ is the estimation value of interpolated point pt_0 , $Z(pt_i)$ the value of the i th sampling point pt_i , and λ_i is the weight coefficient of pt_i indicating the contribution of $Z(pt_i)$ to $Z^*(pt_0)$. The Kriging algorithm uses a semi-variogram to compute the influenced weights of sampled points on the interpolated point in order to realize its attribute estimation. Among all the semi-variograms, the spherical model is commonly used in geology [51] and it can be described as follows:

$$\gamma(h) = \begin{cases} 0 & h = 0 \\ C_0 + C \left(\frac{3}{2} \frac{a}{h} - \frac{1}{2} \frac{h^3}{a^3} \right) & 0 < h \leq a \\ C_0 + C & h > a \end{cases}, \quad (2)$$

where C_0 is the nugget, $C_0 + C$ is the sill value, a is the range and h is the increment of regionalized variables. We denote that $h_{i,j}$ is the distance between pt_i and pt_j . According to an unbiased estimation, Equations (3) formulates the relationship between λ_i and $\gamma(\cdot)$:

$$\begin{aligned} \sum_{j=1}^N \lambda_j \gamma(h_{i,j}) - \mu &= \gamma(h_{i,0}) \\ \sum_{i=1}^N \lambda_i &= 1, \end{aligned} \quad (3)$$

where μ is the Lagrange multiplier and $\gamma(\cdot)$ is the variogram. Thus, Equation (3) can be described as the following matrix style:

$$A\lambda = \gamma_0 \text{ or } \lambda = A^{-1}\gamma_0 \quad (4)$$

$$A = \begin{bmatrix} \gamma_{11} & \gamma_{12} & \cdots & \gamma_{1N} & 1 \\ \gamma_{21} & \gamma_{22} & \cdots & \gamma_{2N} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \gamma_{N1} & \gamma_{N2} & \cdots & \gamma_{NN} & 1 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix}, \lambda = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_N \\ \mu \end{bmatrix}, \gamma_0 = \begin{bmatrix} \gamma_{01} \\ \gamma_{02} \\ \vdots \\ \gamma_{0N} \\ 1 \end{bmatrix}. \quad (5)$$

According to Equations (3) and (4), the solved parameters of the Kriging algorithm is the variogram values in A and λ . For the spherical model, the parameters are generally fixed values that depend on expert experience and have a certain subjectivity. Our study adopts a particle swarm optimization (PSO) algorithm to optimize model parameters, which can reflect the best parameter estimation. PSO was suggested by in a previous work [53] that simulated the swarming behavior of birds in an iterative fashion:

$$v_{i,d}^{t+1} = \omega * v_{i,d}^t + c_1 * rand_1 * (p_{i,d} - x_{i,d}^t) + c_2 * rand_2 * (p_{g,d} - x_{i,d}^t) \quad (6)$$

$$x_{i,d}^{t+1} = x_{i,d}^t + v_{i,d}^{t+1}, \quad (7)$$

where ω is the inertia weight, c_1 and c_2 are learning factors, and $rand_i$ s are random numbers uniformly distributed between [0,1]. This study adopts an ordinary Kriging algorithm and spherical model to interpolate regionalized points. The PSO algorithm optimizes parameters c and C_0 , which are designed as particle position or velocity vector $\langle c, C_0, a \rangle$ with the maximum iterative number $iter_{max} = 1000$ and the threshold value $Value_{threshold} = 10^{-6}$. The fitness function is defined as follows:

$$Fit(pt_j) = |Z^*(pt_j) - Z(pt_j)|, \quad (8)$$

where $Z^*(pt_j)$ is the estimation value and $Z(pt_j)$ is the actual value. For large amounts of data the PSO computation time is too long. Thus, our work adopted GPU to accelerate the convergence speed of the whole swarm and based it on compute unified device architecture (CUDA). The parallel computing model is described in Figure 5.

CUDA is a single instruction multiple data (SIMD) parallel computing model. As a coprocessor, GPU can assist CPU to finish highly parallelized computation task through generating lots of threads. As described in Figure 5, CUDA adopts a multilayer memory architecture including threads, blocks and grids. All the blocks in one grid carry out the same function and do not need to communicate with each other. All of the threads in one block can implement the sharing of data communication by sharing memory and each thread has a private register and local memory. The steps of the GPU-PSO-Kriging algorithm are defined in Algorithm 1.

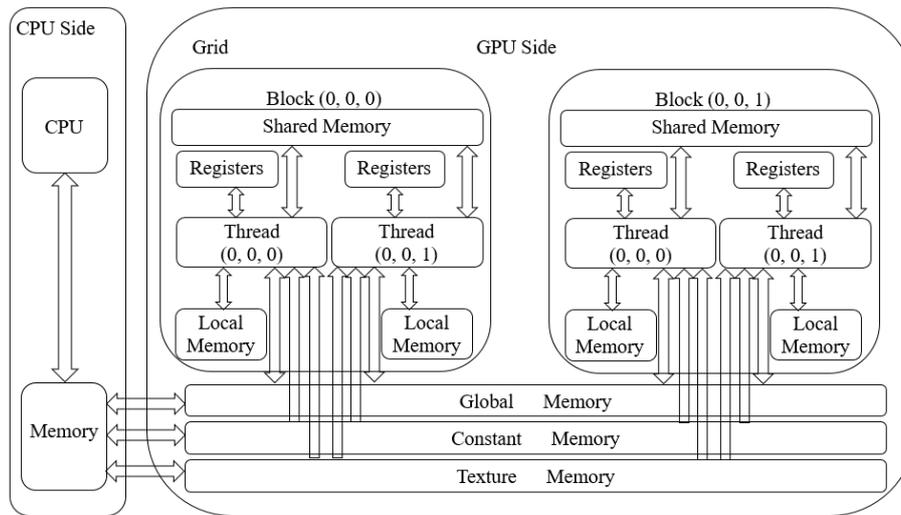


Figure 5. CUDA parallel computing model.

Algorithm 1 The Steps of the GPU-PSO-Kriging Algorithm

Input: $\omega, c1, c2, iter_{max}, Value_{threshold}$

Output: c, C_0, a

Steps

1. Initialize the velocities and positions of all the particles randomly on CPU side; CUP transmits particle information to GPU;
2. Each particle to compute velocity and position by using the Equations (6) and (7) on GPU in parallel;
3. Compute fitness function value by formula (8) to update $p_{i,d}$ on GPU in parallel;
4. On CPU side, update $p_{g,d}$, if reach $iter_{max}$ or $Value_{threshold}$, give optimized c, C_0, a and algorithm finished, else return to step (2).

3.3. Data Modeling

Data modeling is an important process in 3DVF4UDR. As shown in 3DVF4UDR in Figure 6, the interpolated geological data can be sliced at any 3D plane and geological information can be recognized by pickup lines, which can represent closed or non-closed areas. Afterwards, the pickup lines are constructed to different 3D geohazard bodies by using the proposed layer-constrained TIN (LC-TIN) algorithm.

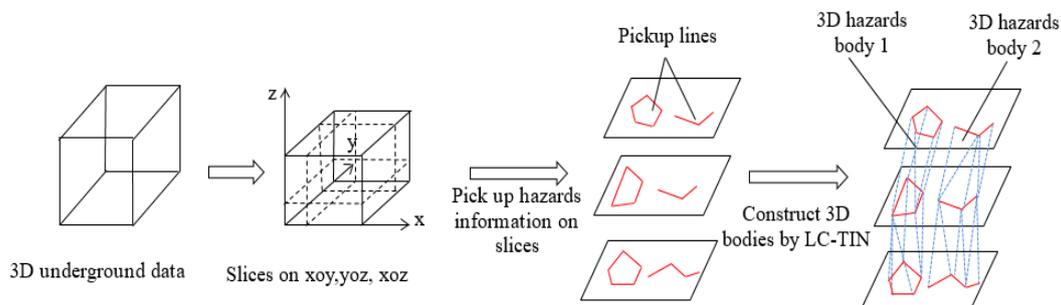


Figure 6. Process of 3D geological data modeling.

The basic conceptions are defined in Table 1 and the LC-TIN algorithm can be described in Algorithm 2 that describes the construction process of geohazard bodies. We assume that the data points are evenly distributed on each layer. It has been seen that given fixed number of total data points, the computing time of LC-TIN decreases as the number of layers, M , increase because the

number of data points on each layer, v , decreases. Thus, the complexity is $O(Mv^2)$ and $M < v$ is the most common case. For unevenly distributed data points, the running time has been tested to be consistent with $O(Mv^2)$ [49].

3.4. Data Analysis

For the 3D geohazard bodies, 3DVF4UDR can analyze their attribute information. This includes the space position computation of geohazard objects, projections to any 2D coordinate planes, and the volume computation of 3D bodies. Data analysis can provide the necessary information for disaster treatment. We adopt the space geometric method to estimate the volumes of geohazard bodies and the whole volume is accumulated by the sub-volumes of η . The steps for computing volume for the b th geohazard body are described in Algorithm 3.

Table 1. Basic conceptions in the LC-TIN algorithm.

Notations	Descriptions
δ	Geological data point $\delta = \langle \zeta, \gamma \rangle$, $\zeta = \langle x, y, z \rangle$ is the 3D coordinate values and γ is the δ attribute value, for example the reflection value of an electromagnetic wave.
ς	Geological data slice, $\varsigma = \begin{pmatrix} \delta_{11} & \cdots & \delta_{1n} \\ \vdots & \ddots & \vdots \\ \delta_{m1} & \cdots & \delta_{mn} \end{pmatrix} = \begin{pmatrix} \delta_1 \\ \cdots \\ \delta_m \end{pmatrix}$, $1 \leq i \leq m, 1 \leq j \leq n$. ς^p is the p th ($p \in [1, u]$) slice.
ℓ	Pickup line: $\ell_p^{b_c}$ is the b_c pickup line on ς^p . $b \in \{1, 2, \dots, t\}$ is the number of the b th geohazard body. b_c is the corresponding c th line of the b th body, $c \in \{1, 2, \dots, N_b\}$ and N_b is the number of pickup lines for the b th body. The δ amount in $\ell_p^{b_c}$ is $N_p^{b_c}$.
ψ	Pickup line set: the ψ of the b th geohazard body is $\psi_b = \{\ell_p^{b_c} p = 1, 2, \dots, u\}$ and $\psi = \{\psi_b b = 1, 2, \dots, t\}$ indicates the whole set of pickup line sets.
η	Adjacent pickup line pair: $\eta_{b_c} = \langle \ell_p^{b_c}, \ell_q^{b_{c+1}} \rangle$ is the η in $\psi_b, c \in \{1, 2, \dots, N_b\}, p, q \in [1, u]$ and $p \neq q$.
ϑ	Base line of η_{b_c} : $\delta_{\vartheta_b}^i$ is the i th δ on ϑ_b that is the base line of $\eta_{b_c}, i \in \{1, 2, \dots, N_{\eta_{b_c}}\}$, $N_{\eta_{b_c}} = \min\{N_p^{b_c}, N_q^{b_{c+1}}\}$. $N_{\eta_{b_c}}$ is the δ number of ϑ_b . non-base line is $\widetilde{\vartheta}_{b_c}$.
σ	Base edge: $\sigma_{\eta_{b_c}}^i$ is the line formed by two adjacent points $\delta_{\vartheta_b}^i$ and $\delta_{\vartheta_b}^{i+1}$ on $\vartheta_b, i \in \{1, 2, \dots, N_{\eta_{b_c}} - 1\}$.
T	Pickup triangle: $T_{\eta_{b_c}}^i = \{\delta_j \delta_j \in \ell_p^{b_c} \text{ or } \ell_q^{b_{c+1}}\}, j \in [1, 3]$. δ_j cannot be on the same ℓ .
Φ	Pickup triangle set: the T set generated by η_{b_c} is $\Phi_{b_c} = \{T_{\eta_{b_c}}^i i \in [1, N_p^{b_c} + N_p^{b_{c+1}} - 2]\}$.
Ω	Pickup 3D body: the 3D body constructed by $\Phi_b = \{\Phi_{b_c} c = 1, 2, \dots, N_b\}$ is named as Ω_b , and $\Omega = \{\Omega_b b = 1, 2, \dots, t\}$.

For road maintenance personnel, the estimation of geohazard volumes and positions can assist them to take necessary countermeasures. When W is large enough, the estimation can reflect the real situation. In Section 3, we have discussed the architecture of 3DVF4UDR that adopts the specification of MVC and all the processes are integrated into one workflow. GPR data is organized as various self-defined files in different workflow steps. Some novel algorithms, which construct the whole chain of underground geohazard recognition, are described in detail. They are the GPU-PSO-Kriging algorithm (in Section 3.2), the LC-TIN algorithm (in Section 3.3) and the space geometric method (in Section 3.4).

Algorithm 2 The Steps of the LC-TIN Algorithm**Input:** ψ **Output:** Ω **Steps**

1. if $\Psi \neq NULL$ {
2. for (b = 1; b < t; b++) {
3. For (c = 1; c < $N_b - 1$; c++) {
4. for (p = 1; p < u; p++) {
5. construct $\eta_{b_c} = \langle \ell_p^{b_c}, \ell_q^{b_{c+1}} \rangle$
6. select ϑ_{b_c} from η_{b_c}
7. for (i = 1; i < $N_{\eta_{b_c}}$; i++) {
8. select $\sigma_{\eta_{b_c}}^i$ from ϑ_{b_c}
9. select the third point $\delta_{\vartheta_{b_c}}^j$ from $\tilde{\vartheta}_{b_c}$ that satisfies $\delta_{\vartheta_{b_c}}^j = \min\{\tau(\delta_{\vartheta_{b_c}}^j) | j = 1, 2, \dots, \max\{N_p^{b_c}, N_q^{b_{c+1}}\}\}$ or
 $\delta_{\vartheta_{b_c}}^j = \max\{\cos(\delta_{\vartheta_{b_c}}^j) | j = 1, 2, \dots, \max\{N_p^{b_c}, N_q^{b_{c+1}}\}\}$
10. construct $T_{\eta_{b_c}}^i$ by $\delta_{\vartheta_{b_c}}^j$ and add it to Φ_{b_c}
11. //end for (7)
12. //end for (4)
13. //end for (3)
14. add Φ_{b_c} to Φ_b
15. //end for (2)
16. generate the whole Φ and form the whole Ω
17. //end if

Algorithms 3 The Steps for Computing a Geohazard Body**Steps**

1. for (p = 1; p < u; p++) {
2. for (c = 1; c < $N_b - 1$; c++) {
3. Construct $\eta_b = \langle \ell_p^{b_c}, \ell_q^{b_{c+1}} \rangle$ and its corresponding $\Phi_{b_c} = \{T_{\eta_{b_c}}^i | i \in \{1, 2, \dots, N_p^{b_c} + N_q^{b_{c+1}} - 2\}\}$, supposing three points of $T_{\eta_{b_c}}^i$ are $\delta_{T_{\eta_{b_c}}^i}^1$, $\delta_{T_{\eta_{b_c}}^i}^2$ and $\delta_{T_{\eta_{b_c}}^i}^3$.
4. Select two points on $\ell_p^{b_c}$ that have the maximum distance (named as Dis_{max}) marked as $r_1^{\ell_p^{b_c}}$ and $r_3^{\ell_p^{b_c}}$.
 Draw circles on ζ regarding $r_1^{\ell_p^{b_c}}$ and $r_3^{\ell_p^{b_c}}$ as circle centers and Dis_{max} as the radius. Two circles intersect at two points, named as $r_2^{\ell_p^{b_c}}$ and $r_4^{\ell_p^{b_c}}$. Construct quadrangles $R_{\ell_p^{b_c}} = \{r_1^{\ell_p^{b_c}}, r_2^{\ell_p^{b_c}}, r_3^{\ell_p^{b_c}}, r_4^{\ell_p^{b_c}}\}$ and
 $R_{\ell_q^{b_{c+1}}} = \{r_1^{\ell_q^{b_{c+1}}}, r_2^{\ell_q^{b_{c+1}}}, r_3^{\ell_q^{b_{c+1}}}, r_4^{\ell_q^{b_{c+1}}}\}$ that can cover most points of $\ell_p^{b_c}$ and $\ell_q^{b_{c+1}}$ respectively.
5. Divide $R_{\ell_p^{b_c}}$ into triangles $T_{\eta_{b_c}}^{p1} = \{r_1^{\ell_p^{b_c}}, r_2^{\ell_p^{b_c}}, r_3^{\ell_p^{b_c}}\}$ and $T_{\eta_{b_c}}^{p2} = \{r_3^{\ell_p^{b_c}}, r_4^{\ell_p^{b_c}}, r_1^{\ell_p^{b_c}}\}$. Divide $R_{\ell_q^{b_{c+1}}}$ into triangles
 $T_{\eta_{b_c}}^{q1} = \{r_1^{\ell_q^{b_{c+1}}}, r_2^{\ell_q^{b_{c+1}}}, r_3^{\ell_q^{b_{c+1}}}\}$ and $T_{\eta_{b_c}}^{q2} = \{r_3^{\ell_q^{b_{c+1}}}, r_4^{\ell_q^{b_{c+1}}}, r_1^{\ell_q^{b_{c+1}}}\}$.
6. Compute $D_{\eta_{b_c}}^1 = \sum_{k=1}^{N_{b_c} + N_{b_{c+1}} - 2} d(T_{\eta_{b_c}}^k)$, where $d(T_{\eta_{b_c}}^k)$ is the absolute value of the mixed product about vectors
 $o\delta_{T_{\eta_{b_c}}^k}^1, o\delta_{T_{\eta_{b_c}}^k}^2$ and $o\delta_{T_{\eta_{b_c}}^k}^3$, where o is the coordinate origin.
7. Compute $D_{\eta_{b_c}}^2 = d(T_{\eta_{b_c}}^{p1}) + d(T_{\eta_{b_c}}^{p2})$ and $D_{\eta_{b_c}}^3 = d(T_{\eta_{b_c}}^{q1}) + d(T_{\eta_{b_c}}^{q2})$. Finally, compute approximate volume of the spatial polyhedron among η_{b_c} , named as $V_{\eta_{b_c}} = D_{\eta_{b_c}}^1 + D_{\eta_{b_c}}^2 + D_{\eta_{b_c}}^3$.
8. //end for (2)
9. //end for (1)
10. Accumulate all the sub-volumes $V_{\eta_{b_c}}$ to get the volume of the b th geohazard body.

4. The Application of 3DVF4UDR

4.1. Experimental Environment and Data

The proposed framework was implemented by C++, OpenGL, CUDA and MFC architecture. The operating system was Windows 10 professional version (64Bit), the software development environment was VS.NET 2019, and the hardware environment was as follows. The CPU was an Intel(R) Core(TM) i7-4800MQ CPU @ 2.70GHz with 16GB memory and the video card was a NVIDIA Quadro K2100M. The experimental data was acquired by our self-developed GPR instrument and the details information is described in Table 2 where twelve sets of samples were drawn. The unit for the coordinates is meter.

Table 2. Acquired experimental data.

No	Start Point (x,y)	End Point (x,y)	Number of Horizontal Positions	Number of Sample Points	Two-Way Travel Time
1	(0,0)	(0,5.5)	392	512	50 ns
2	(0.5,0)	(0.5,5.5)	381	512	50 ns
3	(1,0)	(1,5.5)	381	512	50 ns
4	(1.5,0)	(1.5,5.5)	382	512	50 ns
5	(2,0)	(2,5.5)	384	512	50 ns
6	(2.5,0)	(2.5,5.5)	398	512	50 ns
7	(3,0)	(3,5.5)	377	512	50 ns
8	(3.5,0)	(3.5,5.5)	409	512	50 ns
9	(4,0)	(4,5.5)	396	512	50 ns
10	(4.5,0)	(4.5,5.5)	394	512	50 ns
11	(5,0)	(5,5.5)	382	512	50 ns
12	(5.5,0)	(5.5,5.5)	383	512	50 ns

4.2. Data Preprocessing

As shown in Figure 7, when picking up the data from each survey line and displaying it in the left “DAT AND G3D” window, the corresponding data will be visualized in a “2D view” window. All of the survey line data needs to be preprocessed before further modeling and analyzing.

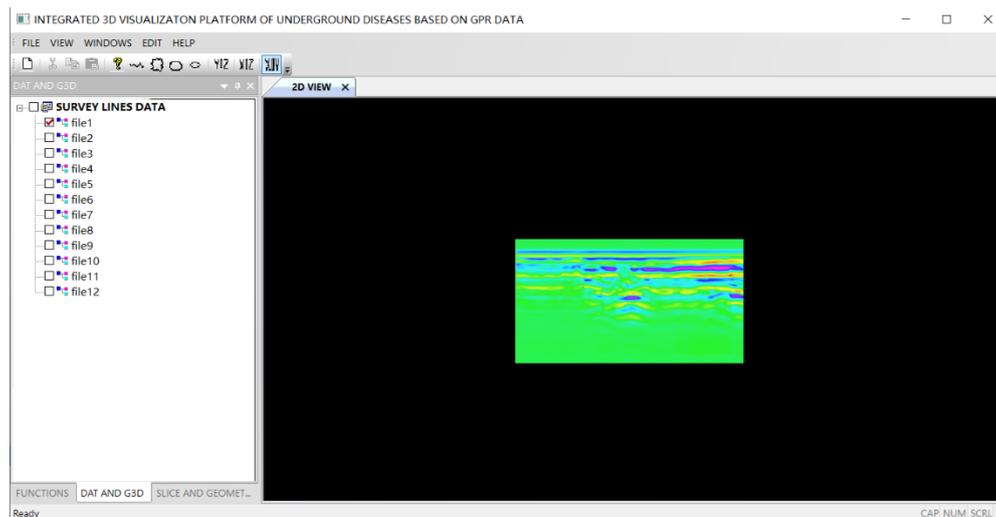


Figure 7. Survey line data visualization.

In Figure 8, all of the partial survey line data (file1.dat~file12.dat) has been adopted to execute the interpolation task. The coordinate ranges of the detection area have been set as required. In this example, a 5.5×5.5 square area has been set with 26 layers in X, Y, and Z directions separately.

The platform configures different Kriging algorithms, such as ordinary Kriging, GPU-PSO-Simple Kriging and GPU-PSO-Ordinary Kriging. Figure 9 shows how their parameters can be set.

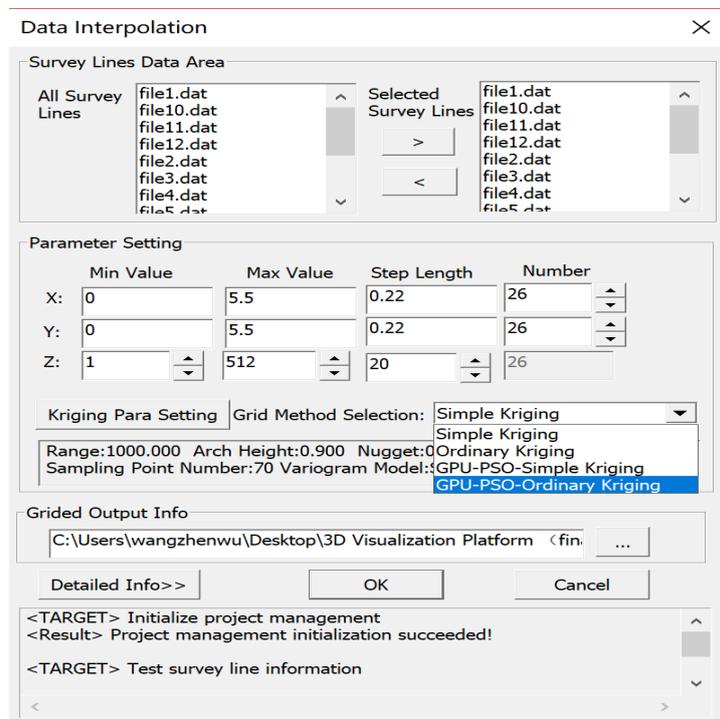


Figure 8. G3D data generation.

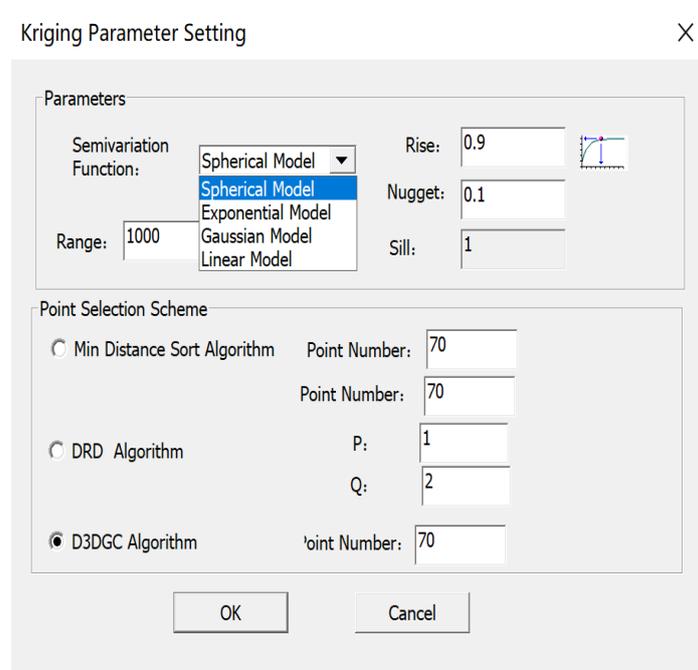


Figure 9. Parameters configuration.

Within the platform it is possible to choose different sampling point selection methods and configure the corresponding parameters. Three selection methods are provided: “Min Distance Sort Algorithm”, “DRD Algorithm” and “D3DGC Algorithm”. All of them have a “Point Number” parameter option while the “DRD Algorithm” also has a “P” and “Q” parameter option. For the

Kriging algorithm itself, some common variogram models have been integrated into the platform, including spherical model, exponential model, Gaussian model and linear model. The platform also sets the initial values of the Kriging parameters, such as nugget, sill, range and rise, but these will be optimized by PSO in GPU-PSO-Kriging algorithm if the latter is chosen (as shown in Figure 9). After generating .G3D data (named K_26_26_26 in Figure 10), the data can be arranged for different layers at any coordinate plane, such as XOY, YOZ and XOZ. It can also be visualized, wholly or partially, and displayed at any angle by dragging the mouse as shown in Figure 10.

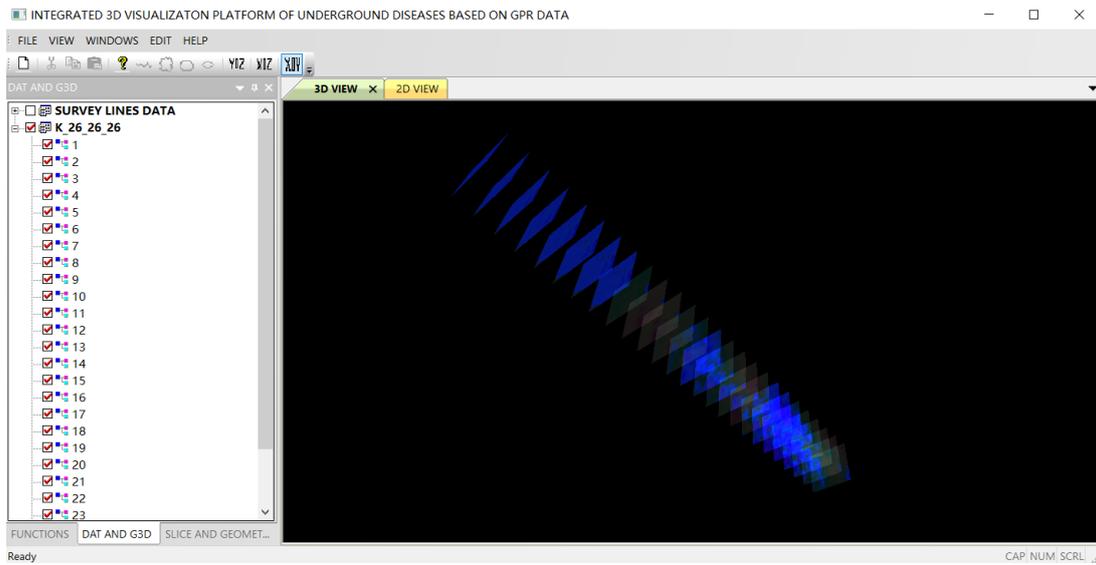


Figure 10. G3D data visualization.

4.3. Data Modeling

As shown in Figure 11, users can pick geohazard information from 26 layers of .G3D data, either in part or as a whole. The layers containing geohazard information can be chosen as slices and the interactive operations can pick up this information on closed or unclosed shapes. After the geohazard information has been selected for all of the slices, the attributes of the different 3D geohazard bodies can be configured. These are organized by pickup lines in different slices including body name, color and texture (Figure 12).



Figure 11. Pick geohazard information.

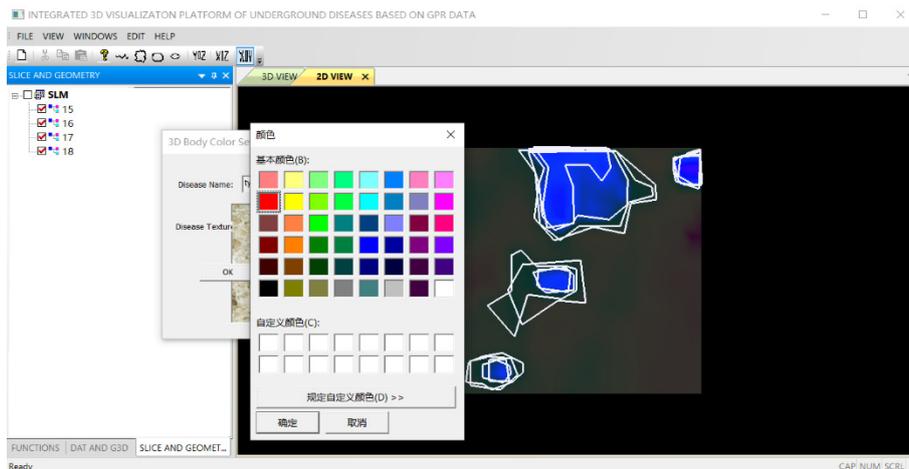


Figure 12. Configuration of attribute information.

In the proposed platform, different Delaunay criteria of the LC-TIN algorithm can be selected in order to implement the construction of the 3D geohazard bodies, where the bodies are assembled by the pickup lines from layer to layer. As shown in Figure 13, when the common criterion Min Area of Circumcircle is specified, a picture of the constructed 3D geohazard bodies displays in the main window.

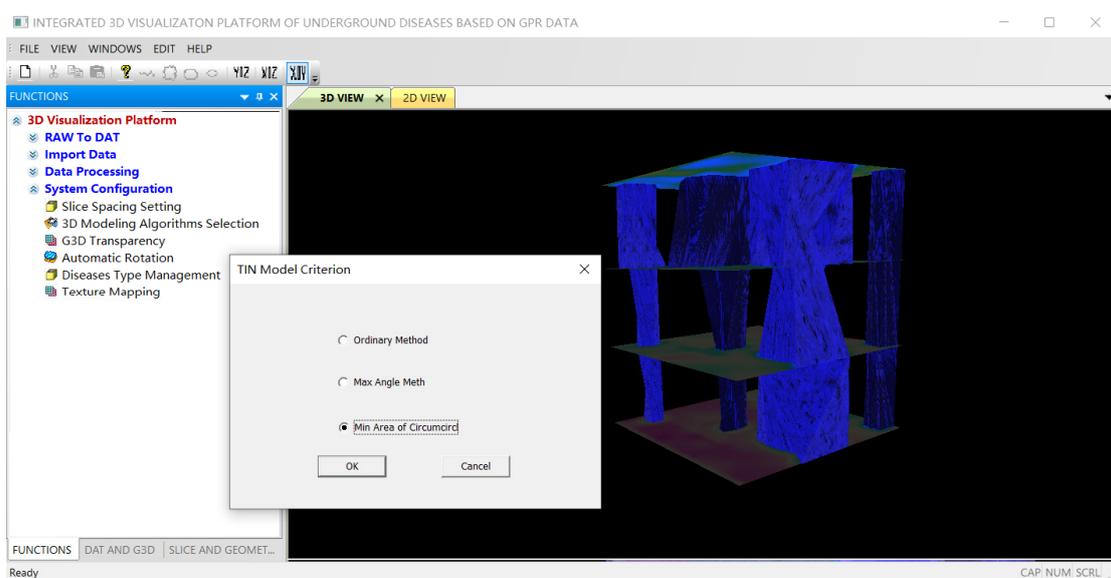


Figure 13. LC-TIN criterion configuration.

4.4. Data Analysis

Geohazard bodies can be visualized and projected to different planes. For example, four geohazard bodies are projected to XOY plane in Figure 14, which can help users to roughly estimate the handled area.

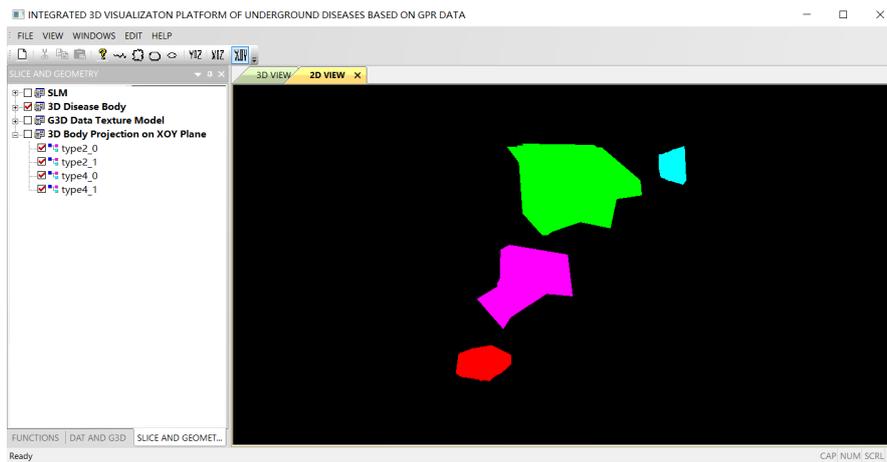


Figure 14. The projection of 3D geohazard bodies.

In Figure 15, the volume or surface area of 3D bodies can be calculated by the space geometric method mentioned in Section 3.4. Users can confirm the geohazard positions and scopes in the detection areas. Moreover, the geohazard bodies can be displayed wholly or partially in 3D scenes through the visualization of those parts between the different slices as shown in Figure 16.

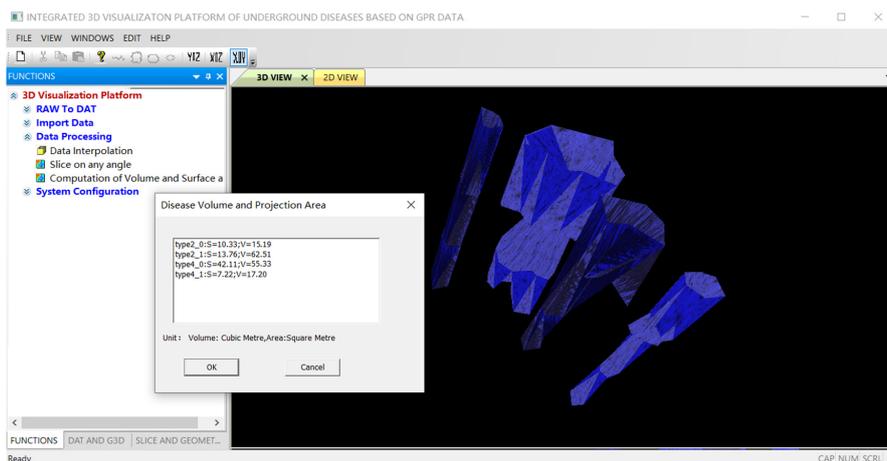


Figure 15. Estimation of volume and surface area.

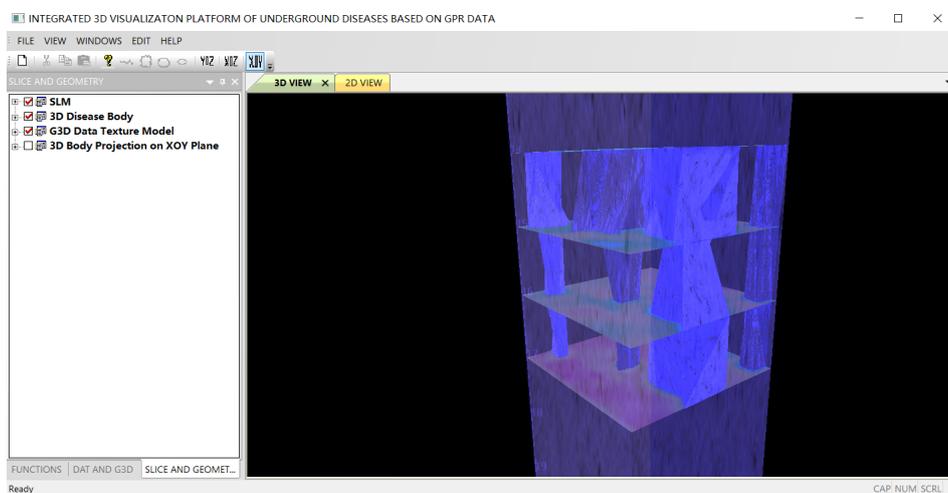


Figure 16. 3D visualization of geohazards.

5. Conclusions and Future Work

In this work, we have presented an interactive 3D visualization platform for underground geohazard recognition on urban roads. A self-developed GPR instrument has been used to acquire the underground road data and 3DVF4UDR provides a workflow to integrate the whole recognition process. In the preprocessing step, two sampling point selection algorithms were adopted in the platform to choose the sampled data more accurately. In order to obtain more detailed underground data, an improved Kriging algorithm was used to interpolate sampling data, which applies the PSO algorithm to optimize the Kriging parameters and employs GPU to enhance the convergence speed. In the modeling step, the LC-TIN algorithm has been used to ensure that the 3D geological bodies display more smoothly and precisely. The space geometry method has been applied to estimate geohazard volumes.

For future works, there are four aspects that can be further explored. (1) The proposed 3DVF4UDR adopts the mode of human-computer interaction to recognize geohazard information, which is based on static GPR data, because it is very difficult to differentiate geohazard character automatically. For example, both pipelines and empty cavities are holes in the ground but only one of them is a geohazard. It is still a challenge to distinguish between them. Future work in this area should look to detect the temporal and spatial changes of empty holes through dynamistic GPR data and parallel computer vision technologies on the assumption that geohazards change form but pipelines do not. (2) In order to detect the change information, a cloud platform could be established and used to store the large amounts of GPR data, which would be necessary to cover the temporal and spatial changes in the same road section. (3) Algorithms charting geological change information could be studied more deeply. Innovative data preprocessing methods, along with the corresponding modeling and analysis technologies, could be studied further so that they can continue to improve the recognition of underground geohazards. (4) More practical comparisons are supposed to be made for further validation and improvement.

Author Contributions: Conceptualization, Zhenwu Wang and Benting Wan; methodology, Zhenwu Wang and Mengjie Han; software, Zhenwu Wang; formal analysis, Zhenwu Wang and Mengjie Han; investigation, Zhenwu Wang, B.W and Mengjie Han; writing—original draft preparation, Zhenwu Wang and Benting Wan; writing—review and editing, Mengjie Han; visualization, Zhenwu Wang; project administration, Zhenwu Wang; funding acquisition, Zhenwu Wang and Benting Wan. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Science Foundation of China, grant 61302157, 61363075 and 51834006; National High Technology Research and Development Program (“863”Program) of China, grant 2012 AA12A308; Yue Qi Young Scholars Program of China University of Mining & Technology, Beijing, grant 8000 15Z1117; Department of Science and Technology of Jiangxi Province of China, grants 20161BBG70078 and KJLD12023; Department of Education of Jiangxi Province of China, grant GJJ180270; and International science and technology cooperation center in Hebei Province, grant 20594501D.

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

References

1. Hadimlioglu, I.A.; King, S.A.; Starek, M.J. FloodSim: Flood Simulation and Visualization Framework Using Position-Based Fluids. *ISPRS Int. J. Geo. Inf.* **2020**, *9*, 163. [[CrossRef](#)]
2. Wielebski, L.; Medyńska-Gulij, B.; Łukasz, H.; Dickmann, F. Time, Spatial, and Descriptive Features of Pedestrian Tracks on Set of Visualizations. *ISPRS Int. J. Geo. Inf.* **2020**, *9*, 348. [[CrossRef](#)]
3. Dalyot, S.; Doytsher, Y. Multi-Temporal Time-Dependent Terrain Visualization through Localized Spatial Correspondence Parameterization. *ISPRS Int. J. Geo. Inf.* **2013**, *2*, 456–479. [[CrossRef](#)]
4. Lathrop, R.; Auermuller, L.; Trimble, J.; Bogner, J. The Application of WebGIS Tools for Visualizing Coastal Flooding Vulnerability and Planning for Resiliency: The New Jersey Experience. *ISPRS Int. J. Geo. Inf.* **2014**, *3*, 408–429. [[CrossRef](#)]
5. Wang, W.; Hu, C.; Chen, N.; Xiao, C.; Jia, S. Spatio-Temporal Risk Assessment Process Modeling for Urban Hazard Events in Sensor Web Environment. *ISPRS Int. J. Geo. Inf.* **2016**, *5*, 203. [[CrossRef](#)]

6. Hu, Y.H.; Bai, Y.C.; Xu, H.J. Analysis of Reasons for Urban Road Collapse and Prevention and Control Countermeasures in Recent Decade of China. *Highway* **2016**, *61*, 130–135.
7. Seong, J.-H.; Jung, M.-H. Determination of priorities for management to reduce collapse accident of open excavation and road sink in urban areas. *J. Korean Tunn. Undergr. Space Assoc.* **2017**, *19*, 489–501. [[CrossRef](#)]
8. Cong, W.; Mao, P.; Zhuang, L. Application of 3D GIS in urban underground space planning. *Chin. J. Geotech. Eng.* **2009**, *31*, 789–792.
9. Evans, R.; Frost, M.; Stonecliffe-Jones, M.; Dixon, N. Ground-penetrating radar investigations for urban roads. *Proc. Inst. Civ. Eng. Munic. Eng.* **2006**, *159*, 105–111. [[CrossRef](#)]
10. Haarder, E.; Looms, M.C.; Jensen, K.H.; Nielsen, L. Visualizing Unsaturated Flow Phenomena Using High-Resolution Reflection Ground Penetrating Radar. *Vadose Zone J.* **2011**, *10*, 84–97. [[CrossRef](#)]
11. Kadioglu, S.; Kadioglu, S. Definition of buried archaeological remains with a new 3D visualization technique of a ground-penetrating radar data set in Temple Augustus in Ankara, Turkey. *Near Surf. Geophys.* **2010**, *8*, 397–406. [[CrossRef](#)]
12. Kadioglu, S.; Kadioglu, Y.K. Visualization of buried anti-tank landmines and soil pollution: Analyses using ground penetrating radar method with attributes and petrographical methods. *Near Surf. Geophys.* **2016**, *14*, 183–195. [[CrossRef](#)]
13. Khwanmuang, W.; Udphuay, S. Ground-penetrating radar attribute analysis for visualization of subsurface archaeological structures. *Geophysicists* **2012**, *31*, 946–949. [[CrossRef](#)]
14. Li, Q.; Yan, Y.; Yang, B.; Hua, X. Research on 3d visualization of underground pipeline. *Editor. Board Geomat. Inf. Sci. Wuhan Univ.* **2003**, *28*, 277–282.
15. Li, X.; Zhu, H. Modeling and Visualization of Underground Structures. *J. Comput. Civ. Eng.* **2009**, *23*, 348–354. [[CrossRef](#)]
16. Munroe, J.S.; Doolittle, J.A.; Kanevskiy, M.Z.; Nelson, F.E.; Hinkel, K.M.; Jones, B.M.; Kimble, J.M. Application of ground-penetrating radar imagery for three-dimensional visualization of near-surface structures in ice-rich permafrost, Barrow, Alaska. *Permafrost. Periglac. Process.* **2007**, *18*, 309–321. [[CrossRef](#)]
17. Tarussov, A.; Vandry, M.; De La Haza, A. Condition assessment of concrete structures using a new analysis method: Ground-penetrating radar computer-assisted visual interpretation. *Constr. Build. Mater.* **2013**, *38*, 1246–1254. [[CrossRef](#)]
18. Varela-González, M.; Solla, M.; Martínez-Sánchez, J.; Arias, P. A semi-automatic processing and visualization tool for ground-penetrating radar pavement thickness data. *Autom. Constr.* **2014**, *45*, 42–49. [[CrossRef](#)]
19. Wang, C.; Chen, J.; Zhu, H.; Li, X. Subterranean pipeline real-time design in three dimensional system. *J. Tongji Univ.* **2008**, *36*, 1332–1336.
20. Wen, F.; Wen, L.; Juan, Y.; Yang, P.; Yu, T. 3D visualization of urban underground pipelines by using SuperMap. In Proceedings of the 5th International Conference on Energy Materials and Environment Engineering, Kuala Lumpur, Malaysia, 12–14 April 2019; pp. 1–9.
21. Hao, M.; Wang, D.; Deng, C.; He, Z.; Zhang, J.; Xue, D.; Ling, X. 3D geological modeling and visualization of above-ground and underground integration taking the Unicorn Island in Tianfu new area as an example. *Earth Sci. Inform.* **2019**, *12*, 465–474. [[CrossRef](#)]
22. Chen, Q.; Liu, G.; Ma, X.; Yao, Z.; Tian, Y.; Wang, H. A virtual globe-based integration and visualization framework for aboveground and underground 3D spatial objects. *Earth Sci. Inform.* **2018**, *11*, 591–603. [[CrossRef](#)]
23. Dinh, K.; Gucunski, N.; Zayed, T. Automated visualization of concrete bridge deck condition from GPR data. *NDT E Int.* **2019**, *102*, 120–128. [[CrossRef](#)]
24. Pereira, M.; Burns, D.; Orfeo, D.; Zhang, Y.; Jiao, L.; Huston, D.; Xia, T. 3-D Multistatic Ground Penetrating Radar Imaging for Augmented Reality Visualization. *IEEE Trans. Geosci. Remote. Sens.* **2020**, *58*, 5666–5675. [[CrossRef](#)]
25. Xu, X.; Peng, S.; Xia, Y.; Ji, W. The development of a multi-channel GPR system for roadbed damage detection. *Microelectron. J.* **2014**, *45*, 1542–1555. [[CrossRef](#)]
26. Liu, Z.; He, X.; Zhang, S. Multi-distance measures weighted Kriging method for deformation stability analysis of steep slopes. *J. Hydraul. Eng.* **2009**, *40*, 709–716.
27. Li, X.; Zhang, Z. A combined indicator-ordinary Kriging method for estimating thicknesses of discontinuous geological strata. *Rock Soil Mech.* **2014**, *35*, 2881–2887.
28. Niu, W.; Zhu, D.; Chen, Q. Bayesian Collocated CoKriging. *J. Comput. Aided Des. Comput. Graph.* **2002**, *14*, 343–347.

29. Yang, D. Spatial prediction using kriging ensemble. *Sol. Energy* **2018**, *171*, 977–982. [[CrossRef](#)]
30. Gribov, A.; Krivoruchko, K. Empirical Bayesian kriging implementation and usage. *Sci. Total. Environ.* **2020**, *722*, 137–290. [[CrossRef](#)]
31. Fang, J.; Gao, Y.; Sun, G.; Xu, C.; Zhang, Y.; Li, Q. Optimization of Spot-Welded Joints Combined Artificial Bee Colony Algorithm with Sequential Kriging Optimization. *Adv. Mech. Eng.* **2015**, *6*, 573–694. [[CrossRef](#)]
32. He, W.; Xu, Y.; Zhou, Y.; Li, Q. A novel improvement of Kriging surrogate model. *Aircr. Eng. Aerosp. Technol.* **2019**, *91*, 994–1001. [[CrossRef](#)]
33. Zhang, G.; Wang, G.; Li, X.; Ren, Y. Global optimization of reliability design for large ball mill gear transmission based on the Kriging model and genetic algorithm. *Mech. Mach. Theory* **2013**, *69*, 321–336. [[CrossRef](#)]
34. Wang, W.; Zhang, R.; Liu, W. Kriging Interpolation Method Optimized by Support Vector Machine and Its Application in Oceanic Data. *Trans. Atmos. Sci.* **2011**, *34*, 567–573.
35. Yan, H.; Li, X.; Wang, S. Reconstruction of three-dimensional temperature field based on least-square method and Kriging interpolation. *J. Shenyang Univ. Technol.* **2014**, *36*, 303–307.
36. Bouhleb, M.A.; Martins, J.R.R.A. Gradient-enhanced kriging for high-dimensional problems. *Eng. Comput.* **2018**, *35*, 157–173. [[CrossRef](#)]
37. Wang, Z.; Bu, Y.; Ma, J. Sampling data selection algorithm for Kriging method based on ground penetrating radar data. *J. Harbin Eng. Univ.* **2017**, *38*, 784–790.
38. Wu, C.L.; Hen, Z.W.; Weng, Z.P.; Liu, J.Q. Property, classification and key technologies of three-dimensional geological data visualization. *Geol. Bull. China* **2011**, *30*, 642–649.
39. Wu, Q.; Xu, H. Research of three-dimensional geological modeling and visualization method. *Sci. China Ser. D Earth Sci.* **2014**, *34*, 54–60.
40. Bi, S.B.; Zhang, G.J.; Hou, R.T.; Liang, J.T. Comparing Research on 3 D Modeling Technology & Its Implement Methods. *J. Wuhan Univ. Technol.* **2010**, *32*, 26–30.
41. Zhu, L. Construction Method and Actualizing Techniques of 3D Visual Model for Geological Faults. *J. Softw.* **2008**, *19*, 2004–2017. [[CrossRef](#)]
42. Liu, H.; Wu, C. Developing a Scene-Based Triangulated Irregular Network (TIN) Technique for Individual Tree Crown Reconstruction with LiDAR Data. *Forests* **2019**, *11*, 28. [[CrossRef](#)]
43. Chen, B.; Fu, Z.; Ouyang, H.; Sun, Y.; Hu, J.; Yan, X.; Zeng, S. High Accuracy Regular Grid Digital Elevation Model Modeling Based on 3D Triangulated Irregular Network Using Total Station Measured Points. *Sens. Lett.* **2014**, *12*, 499–508. [[CrossRef](#)]
44. Quan, Y.; Song, J.; Guo, X.; Miao, Q.; Yang, Y. Filtering LiDAR data based on adjacent triangle of triangulated irregular network. *Multimed. Tools Appl.* **2016**, *76*, 11051–11063. [[CrossRef](#)]
45. Longtin, M.J. Efficient representation of dense elevation grids via triangulated irregular networks. In Proceedings of the Fall Simulation Interoperability Workshop, Orlando, FL, USA, 8–12 September 2014; pp. 162–170.
46. Mao, X.; Zhao, Y.; Tang, Y.; Peng, Z.; Chen, J.; Deng, H. Three-dimensional morphological analysis method for geological interfaces based on tin and its application. *J. Cent. South Univ.* **2013**, *44*, 1493–1499.
47. Wang, L.H.; Gao, X.Z.; Liang, B.B.; Yu, L.; Zhu, X.F. Optimized method of building underwater terrain navigation database based on triangular irregular network. *J. Chin. Inert. Technol.* **2015**, *23*, 345–349.
48. Xuan, H.; Miao, Q.; Liu, R.; Guo, X. A Novel Algorithm Based on Triangulated Irregular Network for Edge Detection from LiDAR Data. *Acta Opt. Sin.* **2014**, *34*, 1228002. [[CrossRef](#)]
49. Wang, Z.; Ma, J. Layer-Constrained Triangulated Irregular Network Algorithm Based on Ground Penetrating Radar Data and Its Application. *J. Beijing Inst. Technol.* **2018**, *27*, 146–154.
50. Liu, Y.S.; Yin, X.Y.; He, W.S. Effects of the factors for spatial correlation analysis on Kriging estimation in reservoir modeling. *J. Univ. Pet. China Nat. Sci. Ed.* **2004**, *28*, 24–27.
51. Deng, M.-G.; Li, W.-C.; Li, B.; Li, L.-H.; Jiang, S.-D.; Xiong, G.-X.; Zhang, X.-S.; Yu, H.-J. Application of Log Kriging on Estimated Reserves of the 10-9 Ore Body of Lutangba in the Gejiu Tin Deposits. *J. China Univ. Min. Technol.* **2007**, *17*, 286–289. [[CrossRef](#)]

52. Reyment, R.A.; Davis, J.C. *Statistics and Data Analysis in Geology*; Wiley: New York, NY, USA, 1988; pp. 1–550.
53. Kennedy, J.; Eberhart, R.C. Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Network, Perth, Australia, 27 November–1 December 1995; pp. 1942–1948.

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



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).