GPU-accelerated fast implementation of shortest path algorithm in the noise simulation analysis system

Yuxiao Qiu1,*, Anxin Zou2, Pengcheng Chen3, Luwen Xu4
1School of Computer Science, Southwest Petroleum University, Chengdu, China
2Electric Power Research Institute, Chongqing Electric Power Company, Chongqing, China
3School of Computer Science, Southwest Petroleum University, Chengdu, China
4Electric Power Research Institute, Chongqing Electric Power Company, Chongqing, China

*Corresponding author e-mail: 934681978@qq.com

Abstract. A graphics processing unit (GPU) framework for the computation of the noise level between the noise source and the receiving point in the noise simulation analysis system is presented. The calculation process of the noise level often encounters a problem of slow convergence, and the calculation amount is large due to problems such as weighted point diffusion, transformation of the potential energy matrix, and gradient degradation. To circumvent these limitations, we devise a GPU-accelerated algorithm to calculate the shortest distance between the noise source and the receiving point, which has been shown to perform 11 times faster than the CPU method. Compare test results between GPU method and CPU method in different mesh density scenarios while maintaining their same calculation accuracy.

1. Introduction

As electricity consumption continues to grow in urban area, power supply is becoming an important issue. In order to meet the demand for electricity, more transformer substations need to be built in the urban areas, whereas the noise generated by substation without effective control will cause environmental problems and influence health of those residents.

As a result, noise control receives much attention during the phases of planning and design, engineering, construction, and operation maintenance of the substations. Especially, simulation calculation of noise is always involved during the whole planning and design phase. Utilizing calculation and analysis of substation noise will not only ensure the project meeting the standard of the emission of environmental noise, but also improve the work efficiency and reduce the engineering cost [1], [2].

Due to the influence of the undulating topography and other factors, the traditional noise optimization methods suffer from various shortcomings in noise simulation calculation of substation. For instance, some commonly used methods such as Dijkstra [3], [4], ant colony algorithm, simulated annealing algorithm [5], [6] and the genetic algorithm can calculate the shortest path in the two-dimensional (2D) space, while failing to guarantee the computational efficiency and accuracy in the three-dimensional (3D) space [7].

In this study, we propose a weighted point spread matrix method based on gradient descent for solving the shortest path problems in the three-dimensional space. The proposed method is able to
obtain the global optimal solution of the shortest path in 3D space with obstacles. Due to the intensive 3D mesh implementation, the preprocessing of the CPU before the single shortest path calculation is huge, and the efficiency is too low in practical applications. Therefore, we apply the GPU acceleration algorithm to improve the calculation speed of the calculation algorithm preprocessing. The algorithm takes time to meet engineering needs.

This paper is organized as follows. In Section II, we introduce the core processing algorithm and the main processing flow for the calculation of noise level. In Section III, we present one optimized GPU-based framework for the proposed algorithm. Numerical results are presented in Section IV. Conclusions and future directions are given in Section V.

2. Methods

2.1. Description of shortest path problem

The core function is to analyze the level of the noise from different receiving points when there are multiple noise sources in the substation environment noise simulation system. Sound propagation obeys a straight line dissemination rule when there is no obstacle between the noise source and the receiver. When the noise happens, sound will bypass obstacles and get to the receiver, and the distance traveled by the noise must be the shortest distance between the noise source and the receiver. If the sound is able to propagate through the obstacle, the shortest distance is considered as a straight line between the noise source and the receiver. This will result in noise attenuation as well. According to the outdoor sound propagation attenuation calculation method and Precise shortest distance value, We can get the total noise value received by the receiver.

![Flow chart of the shortest path method.](image)

Fig. 1 depicts a flow chart of the process of searching for the shortest path in this paper. The global steps can be summarized as follows:

1. To generate a scene diffusion matrix: i) determine the coordinate position of noise source point and obstacle set in the space; ii) take the sound source point as the diffusion center; iii) perform weighted diffusion, traverse the entire scene matrix, and generate the diffusion matrix.

2. To generate a potential energy matrix: generate a potential energy matrix of the same size as a scene matrix.

3. To find the shortest path based on gradient descent method: search the path from the end point along with the direction of gradient descent with a certain step length.

4. To obtain initial solution optimization: optimize the initial shortest path and eliminate errors, search for the inflection points by using the exploration line method, record the coordinate position of the inflection point, and turn the shortest path into the set of inflection points; calculate
the sum of the distances between adjacent inflection points to get the shortest path distance and reduce errors.

(5) To eliminate inflection point error: compare the inflection points with all coordinates of obstacle walls, the closest obstacle coordinate is the corner coordinate that it needs to bypass. Change the inflection coordinates to the corner coordinates. This scheme can eliminate the position error of the inflection point around the obstacle and further reduce the error value of the shortest distance.

Output the shortest distance: The lines between each adjacent inflection points constitutes the shortest path. Calculate the sum of the distances between the inflection points to get the shortest distance.

2.2. Shortest path solution preprocessing method with weighted point spread

Data preprocessing is usually required to implement the method of weighted point diffusion for solving the shortest path problem in a 2D/3D space. The steps for data preprocessing are as follows: In the first step, the diffusion matrix was created with zero initial values, while the value of the starting point was assigned being 1. In the second step, multiplying the diffusion matrix by the weight, then move the diffusion matrix by one unit in a certain direction of the coordinate axis, and then superimpose it on the original diffusion matrix, thus completing a diffusion in a certain direction as shown in Fig. 2, until the diffusion is finished. After completing a complete diffusion, set the coordinate position of the obstacle to 0 to prevent diffusion through the obstacle.

![Diagram of diffusion process in 2D scene](image)

**Figure 2.** The diagram of diffusion process in 2D scene.

Fig. 2 shows a diffusion process in a 2D scene. Under the 3D scene, it increases the two diffusions in the positive and negative directions of the Z-axis.

2.3. Diffusion matrix is transformed into potential energy matrix

In order to get better calculation accuracy, the diffusion matrix is diffused with a smaller weight, resulting in smaller edge elements in the diffusion matrix. In this paper, the potential energy matrix is used to avoid the problem of small diffusion value. The original diffusion matrix simulates the distance relationship between each point in the space and the noise source. The diffusion value of each point is inversely proportional to the distance between each point and the starting points. The relationship is then reversed by the potential energy matrix, that is, the further away from the starting point is, the larger the value is. We want to get the shortest path from each point to the starting point. Just start from the position of the receiving point in the potential energy matrix and find the noise.
source along the direction in which the potential energy value drops the fastest, that is, the gradient falling direction. The path traversed is the shortest path.

Figure 3. 2D energy matrix heat map.

In the 2D matrix scene, the noise source point A with coordinates (40, 40), and the obstacles L1, L2, L3, L4 with coordinates [20, 5: 21] [0: 30, 20] [35: 50, 20] [20:50, 35]. FIG. 3 is a heat map of the potential energy matrix generated when the diffusion is completed.

Figure 4. 2D heat map 3D visualization

In order to more intuitively display the physical meaning of the potential energy matrix, the potential energy matrix is displayed in the form of a 3D graph. As shown in Fig. 4, the potential energy value at the noise source point A is the smallest in the potential energy matrix, similar to the valley bottom. The farther away from the noise source, the larger the potential energy value $f(x, y)$ is. The noise source point and the obstacle position determine the potential energy matrix after the diffusion, so no matter which point you start from, as long as the fastest direction of the slope declines, that is, the negative gradient direction searching for in a certain step size, and the global minimum point, that is, the starting point of the diffusion, can always be found, and the search path is the shortest distance path. If you are looking for the shortest path between point B and the noise source point (point A) in the graph, you only need to start from point B and iterate through the gradient descent direction in a certain step to get the shortest path (White arrow path).

The legend uses the rasterization accuracy of 50*50. In order to reduce the error caused by insufficient rasterization precision when searching for path, the bilinear interpolation method is used to interpolate the potential energy matrix.
2.4. Exploring line method to eliminate the shortest distance error

The length of the shortest path is equal to the search steps, but due to the algorithm characteristics, the limited rasterization accuracy will not completely simulate the true circular diffusion (2D) or spherical diffusion (3D). Therefore, the route without bypassing the obstacle does not iterate along an absolute straight line, and only guarantee iteration around a theoretically correct straight path. The diffusion property prevents the path from passing through the obstacle. It is the shortest path around the obstacle, so the algorithm error will make the experimental result value larger than the theoretical value. The shortest path with no error in the theory of Euclidean coordinate space should be a polyline containing a finite number of inflection points. In order to ensure the shortest path distance is close to the ideal shortest distance, use the exploring line method to find the inflection point in the system. The distance between them is to get the shortest distance, thus eliminating the error that deviates from the ideal straight line propagation path.

![Figure 5. Exploring line method.](image)

The black discrete points in Fig. 5(a) is the set of discrete points of the initial shortest path solution, the rectangle is the obstacle, and the arrows in Fig. 5(b) is the exploring lines. At one end of the shortest path coordinate, the starting or ending point is connected to each subsequent point in sequence, as shown by the arrows in Fig. 5(b) above. Then, to judge whether the exploration line intersects with obstacles. If not, judge the next exploration line. If yes, record this point as the first inflection point. The starting point is then updated to the current inflection point, and the process is repeated and iterated in the direction of the untraversed points. Until traversing to the other end point of the path, all the inflection points have been found. O(n) is the time complexity of the algorithm. The dotted black arrow in Fig. 5(b) points to the first inflection point found.

3. CUDA-BASED FRAMEWORK

CUDA (Compute Unified Device Architecture) is a common parallel computing architecture launched by Nvidia (Nvidia Inc., Santa Clara, CA, USA) [9] to solve complex computational problems by using GPU. Kernel functions are provided by CUDA to perform multithreaded parallel computing.

In the practical noise analysis, it is necessary that dense matrix is used to obtain the shortest path in order to obtain accurate noise values, but it will lead to a high computational cost. CUDA is employed to parallelize our algorithm to balance the efficiency with the accuracy.

Our framework falls into three parts: pre-processing Data, diffusion calculation and gradient descent.

1) Pre-processing data (Kernel 1 to 2).

The required data defined and three-dimensional data is changed into one-dimensional structure, which can ensure data alignment and merge memory access, and improve data reading efficiency. Then, the coordinate positions of all obstacles are read and the appropriate values are assigned to the form of the obstacle matrix. At last, the diffusion matrix is copied to the device (Kernel 1). Because the obstacle matrix needs to be read repeatedly in the calculation process, it is necessary to load the obstacle matrix onto the on-chip memory of GPU to improve the memory reading efficiency. Based on
the data properties of the obstacle matrix, texture memory is used to load this data (Kernel 2). Above all, two kernels are needed to load the data separately.

2) Diffusion calculation (Kernel 3 to 6).
Fist of all, a kernel function is used for multiple threads to access the obstacle matrix simultaneously. (Kernel 3) Secondly, diffusion calculation needs to use a kernel function to improve the efficiency of calculation through multithreading, (Kernel 4) . Finally, two kernel functions are required to determine whether expansion is forbidden , (Kernel 5) ,and to record the area that has been scanned, (Kernel 6) . In the operation of kernel function 5, the current number of diffusions is assigned to him when the value at a certain point is greater than 1. Each point can only be operated once.

3) Gradient descent (Kernel 7 to 8).
A kernel function that bypasses obstacles needs to be defined in advance, (Kernel 7) .In this kernel function, if the value of the obstacle is greater than the surrounding points, the gradient will bypass the obstacle while multiple threads will be processing each point simultaneously. Next, a kernel function is used for gradient descent, (Kernel 8) . The gradient matrix will be calculated in three directions, and each threads will process each point.

4) The data calculated on the gpu are copied back to the host (Kernel 9) .

4. Results
The software environment used was Windows 10 and NVIDIA CUDA v.10.0. The hardware environment comprised a 3.10-GHz Xeon CPU E3-1220 V2 and 16-GB memory, and an Nvidia GeForce GTX TITAN X (Nvidia Corp., Santa Clara, CA, USA) GPU. The GPU card comes along with 80 Stream Multiprocessors, along with a total of 5120 CUDA computing cores along with 12 GB of Memory of memory.

Fig. 6 shows the results from a GPU implementation. The five obstacles with coordinates are (0:24, 15, 0:15) (0:24, 5:15,12) (24, 5:15, 0:15) (12:49, 30, 0:49) (0:35,40, 0:49) the gray barrier walls (M1, M2, M3, M4, M5) are marked in Fig. 6 (a) and (b), as a 3D obstacle scene constructed as an example.
Assume that the point P1 (12, 10, 2) surrounded by the walls is a large transformer in the noise source power station. The noise value received by the other point P2 (21, 48, 48) needs to be obtained by the weighted point spreading algorithm. The calculation can be used to obtain the shortest distance between the two points and then to obtain the noise value received at that point according to the relevant formula of outdoor acoustic propagation characteristics.

The gray paths in Figures 6(c) and (d) are derived from the gradient of the three-dimensional potential energy matrix. Since the diffusion is only an approximate spherical diffusion, the gradient does not give the ideal shortest path. It can be seen that the path is not an absolute straight line when the obstacle is not bypassed. Since the final solution is a numerical solution of the distance value, if the distance value is the product of the number of steps and the step size, the distance value will always be larger because of the error. In order to reduce the error, the optimized path is obtained by exploring the line method.

It can be seen from Fig. 6 (e) and (f) that after optimization, there is the ideal straight path when it is no need to bypass the obstacle, but there is still a certain error between the inflection point and the theoretical correct inflection point position, and the shortest distance value obtained by the experiment is : 89.23m, the theoretical shortest distance value is 86.54m.

According to the calculation principle of outdoor sound propagation, atmospheric sound absorption, the calculation formula of point sound source attenuation is:

\[
\Delta L = 10 \log \left(\frac{1}{4\pi r^2}\right)
\]

### Table 1. Time Comparison Result.

| Sound source point | Receiving point | Result(CPU) | Result(GPU)(m) | CPU(ms) | GPU(ms) | SpeedUp Rate |
|--------------------|----------------|-------------|----------------|---------|---------|--------------|
| Scene A(50*50)     | (12,10,2)      | (21,48,48)  | 108.5654       | 10.503  | 2.868   | 3.662        |
| Scene B(50*50)     | (5,25,5)       | (45,30,35)  | 85.732         | 8.542   | 1.001   | 8.542        |
| Scene A(100*100)   | (24,20,4)      | (42,96,96)  | 214.001        | 161.135 | 34.011  | 4.737        |
| Scene B(100*100)   | (48,40,8)      | (90,60,70)  | 183.753        | 10.503  | 11.891  | 11.856       |

\( \Delta L \) is the attenuation value. \( r \) is the distance from the sound source to the sound receiving point, which is the shortest distance value. Table 2 shows the calculation accuracy of the algorithm. Regardless of the temperature and humidity factors, it is assumed that the noise value of the transformer for the point source is 100 dB. The noise value received by the receiving point is 62.04 dB, and the theoretical received noise value should be 62.31 dB. The calculation only considers the noise propagating along the air, regardless of the noise value of the penetrating obstacle and the noise value of the reflection. The accuracy is greater than 99%, which can meet the engineering needs.

### Table 2. Accuracy Result.

| Theoretical shortest distance (m) | Experimental shortest distance (m) | Theoretical noise reception value (dB) | Actual noise reception value (dB) | Accuracy (%) |
|-----------------------------------|-----------------------------------|---------------------------------------|---------------------------------|--------------|
| 86.54                             | 89.23                             | 62.31                                 | 62.04                           | 99.56        |
Although the above experiment proves the accuracy of the algorithm, the time efficiency of the algorithm consumes 10.5s, so it is difficult to guarantee real-time performance when it is necessary to frequently change the position of the sound source point and the obstacles. In the case of changing the receiving point, it can guarantee good real-time performance, and can dynamically calculate the noise value of the receiving point. Table 1 shows the comparison of performance on CPU and GPU platforms. In the case where the CPU and GPU get exactly the same calculation results, the GPU can provide 11x acceleration.

5. Conclusion

In this paper, we devised the GPU-accelerated algorithm for finding the shortest path which significantly improves the computational efficiency of noise calculation in the substation simulation analysis. Our experiments demonstrate that the accuracy of the original approach optimized by CUDA is the same as the one implemented on CPU while the GPU implementation offers speedup by more than 11 times. This GPU implementation is also flexible to be extended to noise computation of multi noise sources in the same environment. In conclusion, the proposed algorithm is efficient and easy to be implemented on a multi-core CPU or GPU environment. The error of the calculation method for substation noise prediction is less than 1%, which meets the engineering standards.

References

[1] Yi-Ju Tseng, Chuan-En Huang, Chiao-Ni Wen, Po-Yin Lai, Min-Hsien Wu, Yu-Chen Sun, Hsin-Yao Wang, Jang-Jih Lu. Predicting breast cancer metastasis by using serum biomarkers and clinicopathological data with machine learning technologies[J]. International Journal of Medical Informatics, 2019-06-12:79-86

[2] Delahanty Ryan J, Alvarez JoAnn, Flynn Lisa M, Sherwin Robert L, Jones Spencer S. Development and Evaluation of a Machine Learning Model for the Early Identification of Patients at Risk for Sepsis[J]. Annals of emergency medicine, 2019-01-17.

[3] Guo-Jun Qi, Jiebo Luo. Small Data Challenges in Big Data Era: A Survey of Recent Progress on Unsupervised and Semi-Supervised Methods - s[J]. 27 Mar 2019, arXiv:1903.11260v1.

[4] Health and Medicine - Medical Statistics; New Medical Statistics Findings Has Been Reported by Investigators at Diagnostica Stago (A comparison of confidence/credible interval methods for the area under the ROC curve for continuous diagnostic tests with small sample size)[J]. Biotech Business Week, 2018-01-08.

[5] Alok Kumar Dwivedi; Indika Mallawaarachchi; Luis A. Alvarado. Analysis of small sample size studies using nonparametric bootstrap test with pooled resampling method[J]. Statistics in Medicine, 2017: 2187-2205.

[6] Zanetti Miriam Raquel Diniz; Petricelli Carla Dellabarba; Alexandre Sandra Maria; Paschoal Aline; Araujo Jnior Edward; Nakamura Mary Uchiyama. Determination of a cutoff value for pelvic floor distensibility using the Epi-no balloon to predict perineal integrity in vaginal delivery: ROC curve analysis. Prospective observational single cohort study[J]. Sao Paulo medical journal = Revista paulista de medicina, 2016: 97-102.

[7] Guoqiang Yuan, Liang Niu, Yinian Zhang, Xiaoqing Wang, Kejun Ma, Hang Yin, Junqiang Dai, Wangning Zhou, Yawen Pan. Defining optimal cutoff value of MGMT promoter methylation by ROC analysis for clinical setting in glioblastoma patients[J]. Journal of Neuro-Oncology, 2017, Vol.133 (1), pp.193-201

[8] CUDA, C Programming Guide V8. 0, NVIDIA Corp., Santa Clara, CA, USA, Jan. 2017.