

Research Article

Pipeline 3D Modeling Based on High-Definition Rendering Intelligent Calculation

Shao Li  and Yu Zhou

Guangdong Polytechnic of Industry and Commerce, Guangzhou 510510, China

Correspondence should be addressed to Shao Li; 0001190462@gdgm.edu.cn

Received 20 January 2022; Revised 27 February 2022; Accepted 3 March 2022; Published 27 March 2022

Academic Editor: Gengxin Sun

Copyright © 2022 Shao Li and Yu Zhou. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In the processing of panoramic video, projection mapping is a very critical step. The selection of the projection mapping format will affect the performance, transmission mode, and rendering mode of the panoramic video codec. Therefore, this article starts from the projection mapping format, analyzes the mapping process of the standard mapping format, and then proposes a method of rendering panoramic video in the projection mapping format. By analyzing the parallel design schemes of swarm intelligence algorithms under different granularities, this paper proposes a parallel swarm intelligence optimization algorithm design method and then designs and implements a parallel artificial bee colony algorithm. With the help of the ArcGIS Engine development platform, this paper defines the interface for data exchange. With the support of Multipatch format data in ArcGIS software, through secondary development, the three-dimensional pipeline automatic modeling module is established, and the pipeline model is automatically generated. The digital construction and visualization of the company play a driving role. Based on the understanding of the characteristics of the pipeline image itself, combined with the analysis of the shortcomings of the existing methods, this paper proposes a new deep learning-based high-definition rendering solution for the pipeline image. In this paper, the pipeline image is preprocessed, and then the processed pipeline image is converted into a style pipeline image through the pipeline image style transfer technology, and the obtained style pipeline image is postprocessed to enhance the effect. The preprocessing of pipeline images mainly includes pipeline image enhancement and pipeline image filtering operations. Its purpose is to change the distribution of pipeline images to improve the quality of pipeline images and make them more suitable for subsequent style conversion. In the part of pipeline image style conversion, this paper proposes a new deep learning-based pipeline image high-definition rendering network, which consists of three subnetworks: pipeline image feature modeling module, feature model alignment module, and pipeline image re-rendering module. This article has conducted sufficient experiments to fully compare the processing results of the method proposed in this article and other existing methods and at the same time shows the high-quality high-definition rendering results. The experimental results verify the excellent performance of the method proposed in this paper.

1. Introduction

In the social process of sustainable development, the urban underground 3D visualization system is indispensable, and it is also a major way for urban construction, rational use of underground space, and protection of underground pipelines [1]. If we want to make the construction and management of urban underground pipelines faster and more convenient, we must use more modern management methods. In the twenty-first century, in order to have an

accurate judgment on engineering and the scientific nature of engineering projects needs to be strengthened, we should build a more scientific and reasonable urban planning basis. This reduces the “road zipper” that can be seen everywhere and also reduces the potential safety hazards during construction, reduces accidents and disasters caused by urban pipelines, and reduces economic losses. With the continuous development and advancement of science and technology, geographic information system technology, human-computer interaction technology, intelligent underground

detection technology, big data technology, and intelligent management technology have all made huge breakthroughs [2]. Due to the development of the above technology, the city has been promoted. The pipeline project is carried out more quickly. Therefore, urban underground pipeline workers should take the comprehensive application of these advanced technologies to make the exploration and management of urban underground pipelines more convenient, safe, and practical [3].

Traditional urban underground pipeline management methods based on two-dimensional flat maps are difficult to accurately express these complex characteristics of underground pipelines. Therefore, new management methods must be sought for scientific and effective management of urban underground pipelines. Three-dimensional visualization displays complex spatial objects in front of users in a three-dimensional manner. Users can observe the spatial relationship of objects from a full perspective, which can give users a real-world experience and improve the level of scientific management of urban underground pipelines. Compared with the two-dimensional flat map representation, the three-dimensional visualization expression has the characteristics and advantages of strong expressiveness, realistic effects, and clear spatial relationships [4]. The change from two-dimensional to three-dimensional means a change in the way of expressing spatial objects and a deepening of spatial cognition. The use of 3D visualization technology to manage urban underground pipelines and the establishment of a three-dimensional urban underground pipeline GIS that meets actual conditions and actual needs can more intuitively understand the distribution characteristics and spatial relationships of underground pipelines, so as to facilitate the scientific management of underground pipelines [5]. The quality and efficiency of pipeline management work solve the problems of low manual management efficiency and insufficient expressiveness of the two-dimensional pipeline map management method. It provides timely and accurate auxiliary analysis and decision-making for government departments and provides a plan for urban emergencies, so as to truly realize the scientific and modern management of urban underground pipelines and ensure the healthy, sustainable, and long-term development of the city's "lifeline."

Before using H.264/AVC or H.265/HEVC technology to encode panoramic video, the panoramic video needs to be projected from a sphere to a plane and sent to the encoder for processing. When playing a panoramic video, it is necessary to inversely map the panoramic video from the plane back to the spherical surface, so as to restore the stereoscopic effect of the video. Through the analysis and research of swarm intelligence algorithm and combined with the design method of the parallel algorithm, this paper proposes a method to accelerate swarm intelligence algorithm through GPU based on CPU-GPU heterogeneous platform and applies it to artificial bee in swarm intelligence algorithm. Specifically, the technical contributions of this paper can be summarized as follows:

First, based on the MultiPatch 3D digital structure provided by the ArcGIS platform, this paper is developed on

the ArcGIS Engine platform to automatically build a 3D pipeline model based on the measurement data and perform texture mapping. This has achieved good results and provided data support for the establishment of three-dimensional digital cities.

Second, this paper compares and analyzes the existing pipeline image enhancement and pipeline image filtering methods. After comparative analysis, the method suitable for nonphotorealistic cartoon rendering of pipeline images is selected.

Third, in the part of pipeline image style re-rendering, this paper makes an in-depth research and analysis on the existing pipeline image cartoon rendering methods based on deep learning.

Fourth, aiming at some inherent defects in existing methods, a new texture modeling module for multiscale feature fusion is designed. Based on the feature model alignment module of relaxed adaptive instance normalization and the re-rendering module of multiscale feature fusion, an end-to-end style painting rendering network is realized.

2. Related Work

The swarm intelligence algorithm is a very important type of algorithm in the bionic intelligence calculation method. It mainly simulates the social behavior of biological groups. For large-scale and complex optimization problems, the swarm intelligence method can reflect the flexibility and adaptability of its solution [6]. Because the bionic intelligent calculation method does not strictly depend on the mathematical nature of the optimization problem and the application characteristics of the problem itself, it needs to design different fitness functions for different problems, thereby improving the flexibility of algorithm application [7].

Artificial fish swarm algorithm is an efficient bionic intelligent optimization method designed by abstracting fish swarm behavior and applying it to solve optimization problems. It has the advantages of parallelism, globality, and rapidity. Artificial fish is an abstraction of biological fish, including fish foraging, swimming, and other behaviors as well as responses to environmental stimuli. Artificial fish can receive environmental information and give feedback to it and can also further influence other artificial fish individuals through their own behavior. In the artificial fish swarm algorithm, the concept of "field of view" in fish is adopted.

The basic idea of particle swarm optimization is to promote the evolution of the group and find the optimal solution through mutual cooperation and information sharing among particles. The mass and volume of the particles are negligible. Corresponding to complex optimization problems, each particle represents a feasible solution to the problem and can be dynamically optimized by continuously adjusting its position and velocity in the search space.

In foreign countries, many countries, such as the United States, Singapore, Canada, Germany, and France, started early in the research and application of underground

pipeline network management information-rich experience [8]. For example, the integrated pipe network management information system launched is developed directly from the bottom. The system can generate cross-sectional analysis graphics and three-dimensional displays in real time according to user needs [9]. In the ancient city of Paris, France, the underground pipeline network is crisscrossed, but the pipelines that need to be observed can be displayed at will in the urban underground pipeline network management information system established by it, and it is convenient for users to query and operate. The underground integrated pipe network management system developed by the German GEOGRAT company can realize real-time service functions. It has successfully installed 1,600 sets in more than 200 German-speaking cities [10]. The “Philadelphia Model,” a three-dimensional digital model developed by the United States, can be used for immersive roaming in a virtual reality three-dimensional landscape environment through a web browser, and query and analysis can be performed to obtain accurate information on the structure of houses or land. In foreign countries, the availability and use of advanced urban underground management information systems have become an important indicator of the level of urban construction [11].

At present, with the continuous maturity of geographic information system software development technology, more and more domestic companies have independently developed geographic information software, which has sprung up like bamboo shoots after a rain, filling the domestic geographic information market [12]. For example, GeoGlobe, a geographic information service software independently developed by Wuhan Gonow, is a geographic information platform software with a complete structure, powerful functions, flexible deployment, and scalability [13]. China Land Digital takes the self-developed MapGIS software as the basic software development platform. This platform integrates the functional resources accumulated by China Land Digital Group in the implementation of geographic information systems with experts and customers in various fields for more than 20 years [14]. It has realized two- and three-dimensional integrated dynamic management, seamless integration of GIS, and remote sensing image processing platforms [15].

With the development of 3D GIS and the popularization of the industry, many domestic companies have begun to study the 3D visualization of underground pipe networks and have developed specialized business modules for 3D pipe network modeling [16]. Related scholars use the self-developed CityMaker three-dimensional municipal facility management information system platform to define the municipal facility type library (pipe network database) to realize three-dimensional data drive, facility editing, system analysis, and other functions, which can effectively manage underground pipe network facilities [17]. CityMaker 3D municipal facilities management information system supports GIS data, AutoCAD data, field survey form data, and other two-dimensional data-driven generation of 3D municipal facilities data [18]. The user builds a new

underground pipe network facility library in the workspace, extracts the pipe network type definition and related rules that need to be processed from the pipe network type library, and uses the system to provide two-dimensional data import and data processing tools to import user data into the system [19]. The underground pipeline network data is driven by two-dimensional data to generate three-dimensional data [20].

3. Method

3.1. Coordinate System in the Process of Panoramic Pipeline Projection. This article uses a coordinate system to represent the geometric position of a point in space. The center of the sphere is at the origin of the Cartesian coordinate system, and the radius of the sphere is 1. For a certain point on the sphere, you can use latitude and longitude (φ, θ) to describe its position. The latitude and longitude lines passing through various points on the sphere form a grid of latitude and longitude maps. f is the longitude of this point, defined as the angle between the line between the projection point of this point on the xz plane and the center of the sphere and the positive x -axis in the counterclockwise direction. θ is the latitude of this point, defined as the angle between the line between this point and the center of the sphere and the xz plane. The range of longitude is $[-\pi, \pi]$, and the range of latitude is $[-\pi/2, \pi/2]$. If the longitude and latitude (φ, θ) of a point are known, then its coordinates in the three-dimensional rectangular coordinate system can be calculated by the following formula:

$$\begin{cases} x = \sin(\theta) \cdot \sin(\varphi), \\ y = 1 - \cos(\theta), \\ z = 1 - \sin(\theta) \cdot \cos(\varphi). \end{cases} \quad (1)$$

Conversely, if the coordinates $(x, y, \text{ and } z)$ of a point on a spherical surface in a three-dimensional rectangular coordinate system are known, then its latitude and longitude (φ, θ) can be calculated as follows:

$$\begin{aligned} \varphi &= -\arcsin\left(\frac{z}{x}\right), \\ \theta &= -\arcsin\left(\frac{x}{\sqrt{x^2 + y^2 + z^2}}\right). \end{aligned} \quad (2)$$

In the process of mapping the sphere to the plane, some mapping formats will map the sphere onto one plane, and some mapping formats will map the sphere onto multiple planes. In order to distinguish the sampling points on multiple projection planes, each projection plane needs to be numbered, denoted by f . For example, the ERP format maps the spherical surface to a plane, and the number of the plane is 0; CMP, EAC, ACP, and other formats map the spherical surface to six planes, and the numbers of the six surfaces are 0, 1, 2, 3, 4, and 5. For each mapping surface, set its resolution to $W \times H$ and use the upper left corner of the mapping surface as the origin of the coordinates to establish a rectangular coordinate system. Then the position of each pixel in

the mapping surface can be in two-dimensional coordinates (m, n) , where $m \in [0, W)$, $n \in [0, H)$; the mapping plane can also be called (m, n) plane.

In order to facilitate the description of the relative positions of the pixels on the projection plane and exclude the influence of the size of the projection plane itself, the pixel coordinates of the pixels can be converted to (u, v) coordinates. The conversion relationship between (u, V) coordinates and (m, n) coordinates is as follows:

$$\begin{aligned} u &= \frac{(m + 0.5)}{W} m \longrightarrow (-1, W), \\ v &= \frac{(n + 0.5)}{(1 - H)} n \longrightarrow (-1, H), \end{aligned} \quad (3)$$

where $u [0, 1)$ and $v [0, 1)$. It should be noted here that each mapping format will translate the coordinates of the (m, n) plane by 0.5 before scaling. The coordinates of the (m, n) plane will be in the range of $[0, W-1]$, so that the coordinates of the (m, n) plane are evenly distributed before being mapped to the unit plane.

3.2. Panoramic Video Rendering Based on Projection Mapping Format. The spatial three-dimensional coordinates determine the relative position of the point in space, and the texture coordinates determine the correspondence between the pixel color of the point and the texture pipeline image. Since ERP is a cylindrical equidistant projection when mapping from the spherical surface to the projection plane, the meridians are mapped into parallel equidistant straight lines in the vertical direction, and the latitudes are mapped into parallel equidistant straight lines in the horizontal direction. If the longitude and latitude lines on the projection surface are evenly spaced and evenly spaced and the projection surface is cut into multiple grids, the texture coordinates (u, v) can be calculated according to the proportional relationship. After calculating (φ, θ) , the three-dimensional space coordinates (x, y, z) of the corresponding spherical grid point can be obtained.

When dividing the grid, the spacing between the warp lines will affect the number of grid points. In this paper, the number of grids divided in the horizontal direction is defined as the number of faces of the sphere grid model, denoted by np because the spacing between the warp lines and the spacing between the latitude lines are the same, and the aspect ratio of the ERP format is 2:1, so the number of patches in the vertical direction is $np/2$. Therefore, the number of grid points in the horizontal direction is $np + 1$, and the number of grid points in the vertical direction is $np/2 + 1$. The entire ERP video frame is divided into $np \times (np/2 + 1)$ nets. grid. Since there is a total of 2π in the horizontal direction and a total of p in the vertical direction, for the (i, j) -th grid point on the ERP, the latitude and longitude are, respectively, as follows:

$$\begin{aligned} \theta &= \frac{2\pi i}{1 - np} i \longrightarrow (-1, np - 1), \\ \phi &= \frac{2\pi j}{np - 1} j \longrightarrow (-1, 2np - 1). \end{aligned} \quad (4)$$

After determining the latitude and longitude of the point, the spatial rectangular coordinates of the grid point can be obtained, which are as follows:

$$\begin{cases} x = \sin(\theta) \cdot \cos(\phi), \\ y = 1 - \cos(\phi), \\ z = 1 - |\cos(\theta) \cdot \sin(2\phi)|. \end{cases} \quad (5)$$

Since the range of texture coordinates is $[0, 1]$, the texture coordinates of this point are as follows:

$$\begin{aligned} u &= \frac{\pi i}{np - 1}, \\ v &= \frac{2j}{np - 2}. \end{aligned} \quad (6)$$

When the video resolution is constant, the larger the distance between the warp and latitude lines, the smaller the number of grids divided, and the larger the distance between the warp lines, the greater the number of grids divided. Within a certain range, the greater the number of mesh vertices, the finer the sampling of texture data and the better the rendering quality; when the number of vertices exceeds this range, the rendering quality cannot be increased, but the rendering performance decreases.

After obtaining the space coordinates and texture coordinates of each grid point of the sphere grid model, then the vertex information needs to be assigned to OpenGL. In this process, OpenGL requires the vertex information to be stored in a way related to the drawing method. The drawing methods in OpenGL are points, triangles, quadrilaterals, polygons, and so on. Since triangles can form quadrilaterals and polygons of any shape, triangles are the most commonly used drawing method.

3.3. Inherent Parallelism of Swarm Intelligence Algorithms. Swarm intelligence algorithms have inherent parallelism. Although the social behavior and evolutionary mode of the group are carried out in the unit of the group, the behavior mode of the individuals in the group has a certain degree of independence. In the optimization process, the swarm intelligence algorithm searches for the optimal solution in the solution space through multiple iterations. The search mechanism is that as the iterative process progresses, the group tends to approach the global optimal solution. Even if the current individual falls into the local optimal solution in the search process, other individuals can gradually jump out of the local optimal by exchanging information with the current individual. Each iteration includes the implementation of social behaviors in the group (there may be multiple behaviors) and the exchange of information

between individuals. In each iteration, the group's social behavior and information exchange usually alternate.

When the group conducts social behavior, the individual searches randomly in the space around him to obtain the maximum benefit. Therefore, there is implicit parallelism between individuals. If it is done on the CPU side, individuals will search for iterations in serial order. If it is on the GPU, a more efficient search method can be designed to allow multiple individuals to search the solution space in parallel at the same time, thereby greatly improving the search efficiency. In addition, the key to information exchange and sharing between individuals is to record individual historical information and global optimal information. Obtaining the global optimal information on the CPU side generally requires a time complexity of (N) . If the parallel programming model of GPU is used, the time complexity can be reduced to $(\log N)$. In addition, the calculation of the fitness of an individual's location often takes a lot of time, and with the expansion of individual dimensions and population size, the computational complexity is greatly increased. In this process, GPU parallel computing can be used to quickly obtain individual fitness. In summary, the key steps in the swarm intelligence algorithm can all be designed in parallel.

3.4. Parallel Program Design of Swarm Intelligence Algorithm.

GPUs are designed based on a streaming multiprocessor architecture, with very limited on-chip computing resources. As a rule of thumb, the speedup will be optimal when the number of thread blocks is twice the number of streaming multiprocessors in the GPU. If it is designed according to a fine-grained scheme, when the population size of the algorithm is large, hundreds or thousands of thread blocks will be concurrently performed, and the number of GPU stream multiprocessors is relatively small (e.g., the Fermi architecture has 16 stream multiprocessors), which causes a large number of thread blocks to block on the chip, which in turn leads to a decrease in execution efficiency. Therefore, implementing a fine-grained scheme will reduce the execution efficiency of the algorithm.

In a coarse-grained design scheme, one thread can be used to correspond to one individual in the group, that is, one thread block processes multiple individuals. Suppose the number of individuals is N , the dimension of individuals is D , and each thread block starts M threads. Then the kernel function needs to start $(N + M - 1)/M$ thread blocks in total. In the coarse-grained scheme, all thread blocks are processed in parallel on the GPU, and all threads in the thread block are also executed concurrently. Since each thread processes an individual, the thread processes the D dimensions of the individual sequentially.

Although the coarse-grained scheme is theoretically less parallel than the fine-grained scheme, it is very efficient in actual design and implementation. First of all, because the CUDA compiler will optimize the loop accordingly, when a thread processes multiple dimensions of an individual, its execution efficiency will not be significantly reduced. Second, in this solution, the number of threads in the thread

block and the number of thread blocks in the thread grid can be flexibly set. Set different M according to different population sizes to achieve the best optimization efficiency.

Figure 1 shows the overall process of swarm intelligence algorithm parallelization. The parallelized swarm intelligence algorithm is still consistent with the flow of the serial algorithm. It includes functional modules such as group initialization, the implementation of group social behaviors, and information exchange between individuals. The group initialization is performed only once at the beginning of the algorithm, and the random number generator is used to initialize the individual position and use it as the input of the subsequent functional modules. Since there is no exchange and cooperation information between the module particles, it can be implemented in parallel. In each iteration, there is a data dependency relationship between the implementation of group social behavior and information exchange and other functional modules, that is, each module must rely on the data flow of the forward module, so the functional modules must be executed in a certain order. In the implementation of group social behavior, because there is less interaction between individuals, this part is easy to parallelize. In the information exchange phase, the process needs to be re-designed according to the programming characteristics of the GPU, so as to efficiently obtain effective information in the group and use it for the implementation of the next module. After completing multiple iterations and satisfying the termination condition, the algorithm will obtain a relatively optimal solution to the problem.

3.5. *Parallel Artificial Bee Colony Algorithm.* Artificial bee colony algorithm is a bionic intelligent calculation method that simulates the process of searching for nectar by bees in different divisions of labor in the colony. Although the hierarchical and well-defined bees in the bee colony can only complete a single task, the bees of different divisions of labor can exchange information through various methods such as swing dance and smell so that the bee colony can quickly find and mine excellent source of nectar.

Artificial bee colony algorithm mainly includes three stages: picking bees, following bees, and reconnaissance bees. The parallel artificial bee colony algorithm is designed according to the method of the parallel swarm intelligence algorithm. The overall flow of the algorithm is shown in Figure 2.

3.6. Geometric Objects and Reference Spaces in ArcGIS.

The main object model contained in geometry: Point is a 0-dimensional geometric figure with X and Y coordinate values; MultiPoint point set is a collection of a series of irregular points, these points contain the same attribute information. The geometric object is defined by the end point and the parameter. Path is a collection of continuous segments, except for the first segment and the last segment in the path; the starting points of the remaining segments are the end points of the previous segment, that is, the segment in the path object does not appear. Ring is a closed path, that is, the start point and the end point have the same

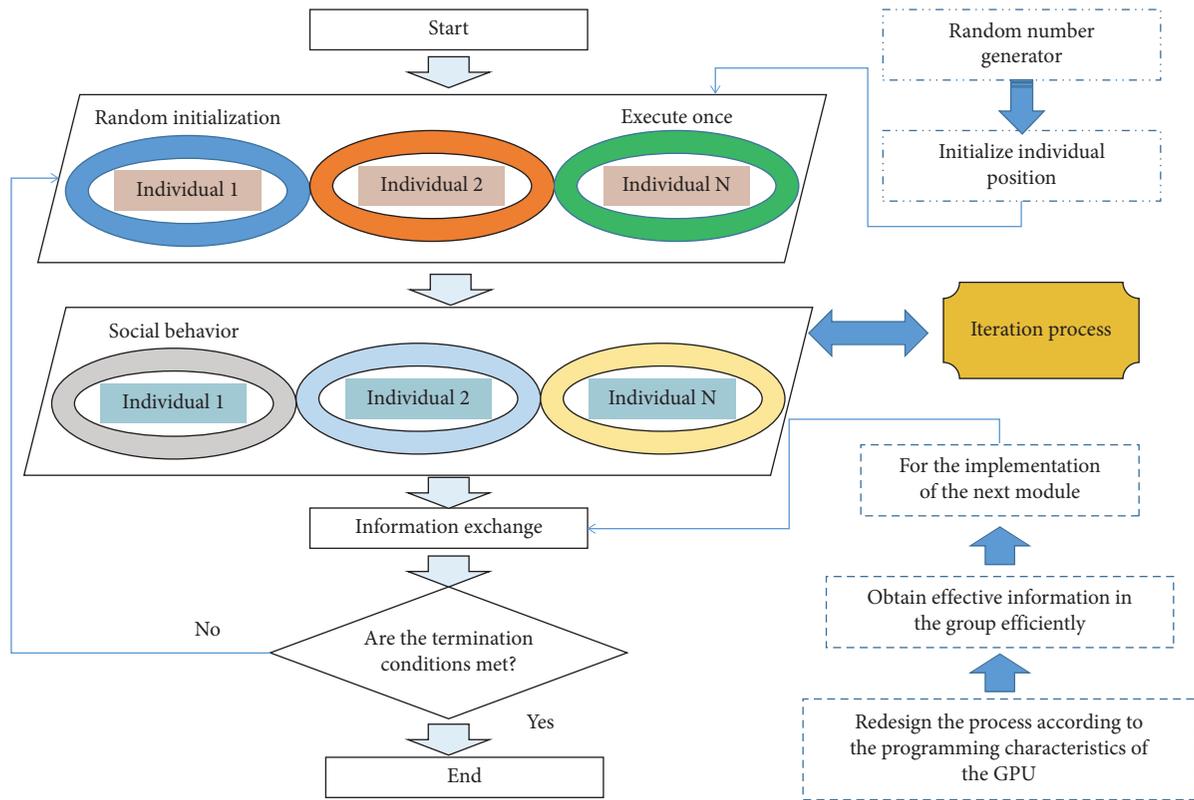


FIGURE 1: The overall process of group intelligence algorithm parallelization.

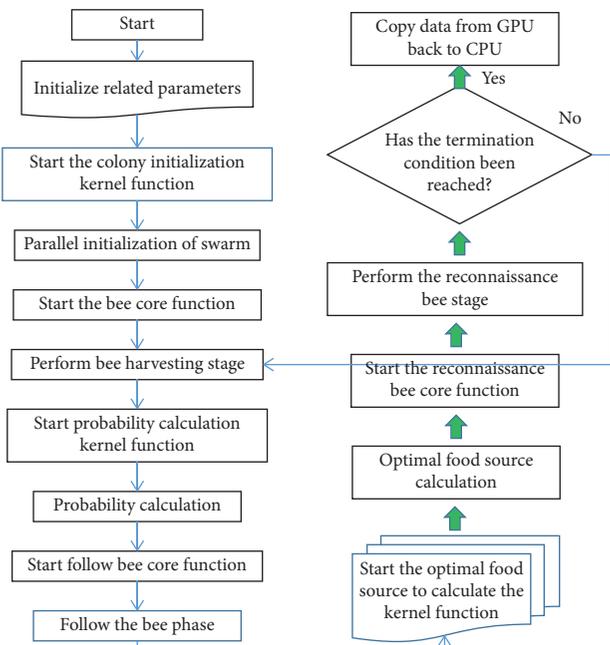


FIGURE 2: Parallel artificial bee colony algorithm.

coordinates, and it has internal and external attributes; and Polyline geometry object is an ordered collection of one or more connected or disconnected path objects. It can be composed of a single path object, multiple connected path objects, or multiple separated path objects. Envelope is a

rectangle that circumscribes all geometric objects and is used to represent the smallest border outside the geometric object. There is an Envelope object; except for Point, Multi-Point, and Envelope, all other geometric objects can be regarded as Curve geometric objects; Multipatch geometric objects are used to describe three-dimensional graphics.

3.7. Model Construction in 3D Software. In ArcGIS software, the multislice data structure provides new ideas for the establishment of 3D models. The three-dimensional model constructed by using Multipatch data type can be stored as a vector feature in a spatial data table of Multipatch type.

Multipatch data structure is a data structure that exists in parallel with points, lines, and areas in ArcGIS data structures. It is a very central construct in ArcGIS and the basis for describing regular or irregular three-dimensional entities on the ArcGIS platform. For some relatively regular models, other geographic information system (GIS) software can also represent, such as buildings and roads. But, for irregular 3D models, node-level characterization of 3D models is required. In this regard, ArcGIS has shown its advantages, and it can use Multipatch to achieve node-level generation, management, editing, and analysis.

With ArcGIS, node-level operations can be done, it means that each node can be inquired, modified, and generated. In the actual use process, a large number of three-dimensional nodes need to be generated, but it does not mean that each node needs to be manually calculated and then added.

4. Simulation Results and Analysis

4.1. Preprocessing Results. Figure 3 shows the effect of pipeline image enhancement processing. It can be seen that after the pipeline image enhancement processing, the color dynamic range of the pipeline image is increased, and the overall picture looks bright and gorgeous. At the same time, the edge features in the pipeline image have been better maintained. The higher dynamic range of pipeline images is conducive to the subsequent high-definition rendering to produce colorful and gorgeous pictures, which can effectively improve the quality of high-definition rendering.

Figure 4 further shows the pipeline image histogram distribution before and after the pipeline image enhancement operation. It can be clearly seen from the figure that the pipeline image enhancement here has an effect similar to histogram equalization. After the pipeline image enhancement, the pixel values of each channel tend to be evenly distributed.

4.2. Model Training Process. Figure 5 shows the change curve of the total generator loss value and the total discriminator loss value as the number of iterations increases during the training process. It can be seen that in the initial stage of training, both the generation loss and the discrimination loss are relatively stable, indicating that the training of the generative adversarial network is close to the Nash equilibrium state, and the discriminator can already distinguish the authenticity of the generated results.

Shortly after the start of training, the generator hardly learned anything about style. For ease of understanding, sometimes, the input pipeline image is called a content map because this part of the pipeline image provides the semantic content of the result. Motivated by reconstruction loss items and counter loss items, the generator can still roughly reconstruct the semantic content in the content map, but it can be clearly observed that the rendering results at the initial training stage do not show obvious pipeline image style. The results all show a dim effect. This is because the network has not learned how to characterize the style and how to insert this style into the content map through the normalization operation of the relaxed pose adaptive instance in the feature alignment module. The feature model alignment module will try to insert the average “style” sampled from the normal distribution into the feature model of the content map, so the rendering result at this time will appear similar to an average and fuzzy style. It can be seen that with the continuous training of the network, the network has gradually learned how to use the feature model modeling module to extract the features that best characterize the pipeline image style and inject this captured style into the feature model of the input content map. At the 400,000th iteration, the high-definition rendering network can already give the input content a distinct style. Compared with high-definition rendering tasks, pipeline image reconstruction tasks are significantly easier to learn. Soon after the start of training, the rendering network designed in this paper can quickly realize the pipeline image reconstruction function. The

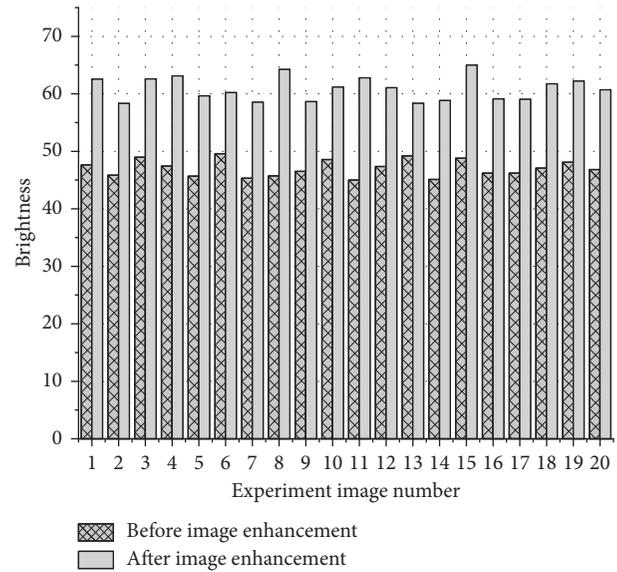


FIGURE 3: Display of pipeline image filtering effect.

naked eye can hardly distinguish the difference between the original pipeline image and the reconstructed pipeline image. Table 1 shows the changes in the quality of the reconstructed content pipeline image with the number of iterations. It can be seen that the rendering network can basically reconstruct the original pipeline image after 7,500 iterations.

4.3. Results Display and Comparison. In order to verify the generalization ability of the trained model, this paper collected 2,000 pipeline images as the test data set during the test, and the resolution of the pipeline image has been ranged from $300 * 300$ to $5,000 * 5,000$. The ultrahigh-resolution pipeline images are all reflected in the test data set. The model used in the test has undergone 300,000 iterations. The final training model size of the multiscale feature modeling network is about 26 Mb, and the model size of the feature model alignment module and the pipeline image re-rendering module is about 12 Mb in total. When testing, first, we preprocess the pipeline image used for testing. Among them, the parameters of the guided filter are set as the filter radius $r=9$ and the regular coefficient $\text{eps}=100$.

The rendering network in this article is based on the implementation of examples. It receives two inputs, the input pipeline image, and the reference pipeline image. Figure 6 shows some processing results of the high-definition rendering network in this article. It can be seen that the method in this article can produce good results for different pipeline image input and style input. The color of the pipeline image after rendering becomes rich and full, and it tends to be smooth in local areas. At the same time, there is almost no obvious (A)rtifacts in the rendering results. The global shape and local detail texture of the objects in the figure are well maintained. It can be said that the changes before and after rendering are limited to the pipeline image.

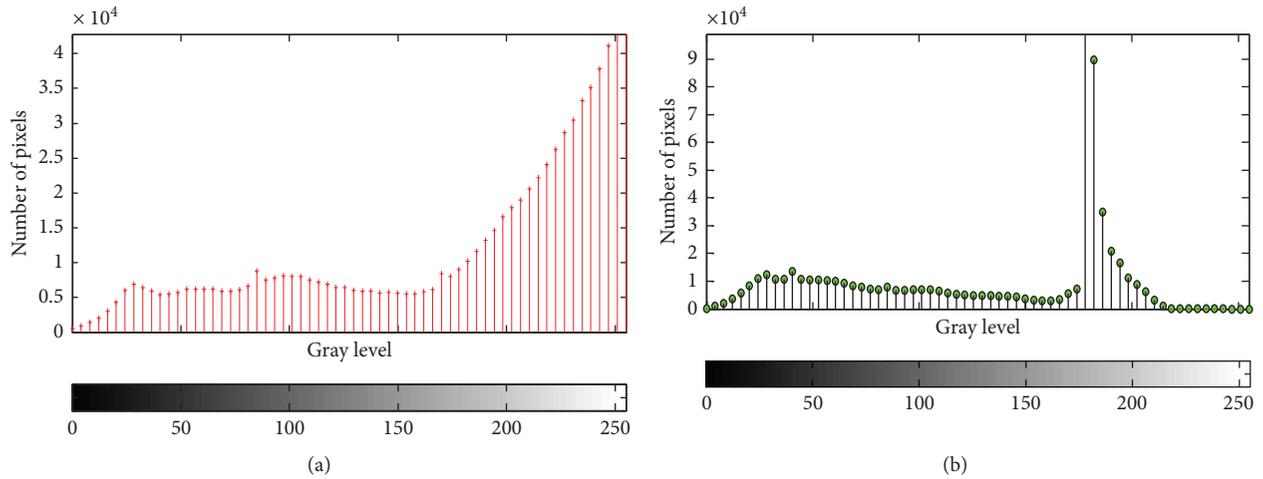


FIGURE 4: Comparison of histograms before and after pipeline image enhancement: (a) before enhancement and (b) after enhancement.

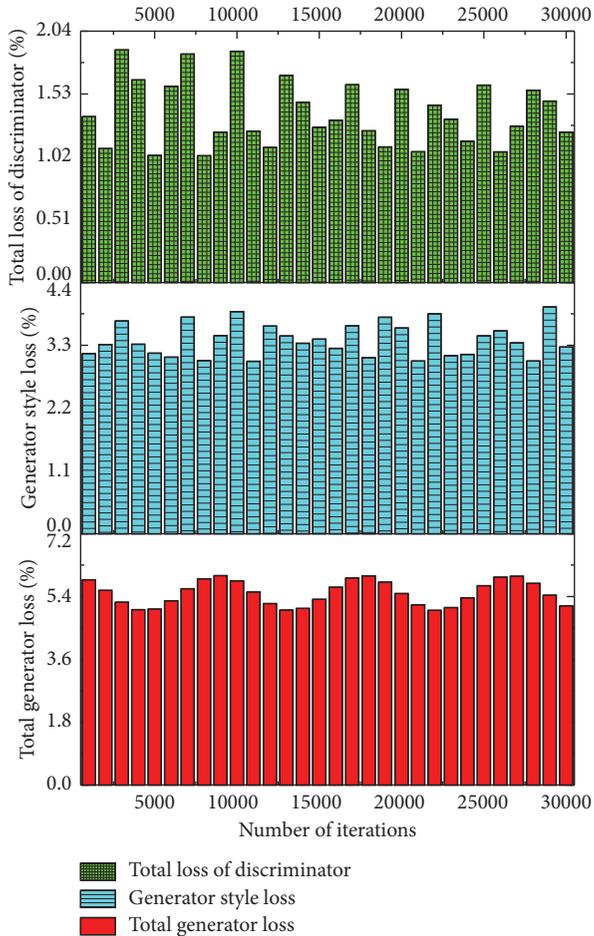


FIGURE 5: The change of the loss value during the training process with the number of iterations: (a) total generator loss, (b) generator style loss item, and (c) total discriminator loss.

In order to visually demonstrate the advantages of the method in this article over other existing methods, we trained the models of the other three methods on the same

TABLE 1: The quality of pipeline image reconstruction changes with the number of iterations.

Number of iterations	7,500	15,000	22,500	30,000
SSIM	0.78	0.81	0.83	0.86
PSNR	21	24	25	27

training data, namely, the CartoonGAN model, AdaIN model, and CycleGAN model. Table 2 lists the model size and the number of parameters of each model. The model parameters of the AdaIN method and the parallel artificial bee colony algorithm are composed of two parts. AdaIN is the encoding network and the decoding network. The method in this article is feature modeling. Network and feature alignment and pipeline image re-rendering network are two parts. Among them, each column represents the results of different methods for the same input. Note that both AdaIN and the method in this article are based on examples, so two inputs are required. The corresponding three-dimensional pipeline model is displayed in the lower right corner of the corresponding result diagram.

Our method accomplishes the task of high-definition rendering well, and the visual effect of the rendering result is significantly better than other methods. In some examples, CycleGAN's output results will produce very serious artifacts (Artifacts). In other examples, CycleGAN's results are not much different from the input pipeline image. The result of the AdaIN model has certain style characteristics in style, but it produces very serious artifacts, and the global style characteristics are also inconsistent. Each part reflects a different color distribution. This is because the AdaIN method is only in a single dimension. The statistical features of the feature map are aligned, and coarse adaptive instance normalization is used. As discussed in the previous article, the adaptive instance normalization will destroy the global consistency of the feature map. The output result of CartoonGAN well retains the semantic content of the original pipeline image, but the style of the result is not

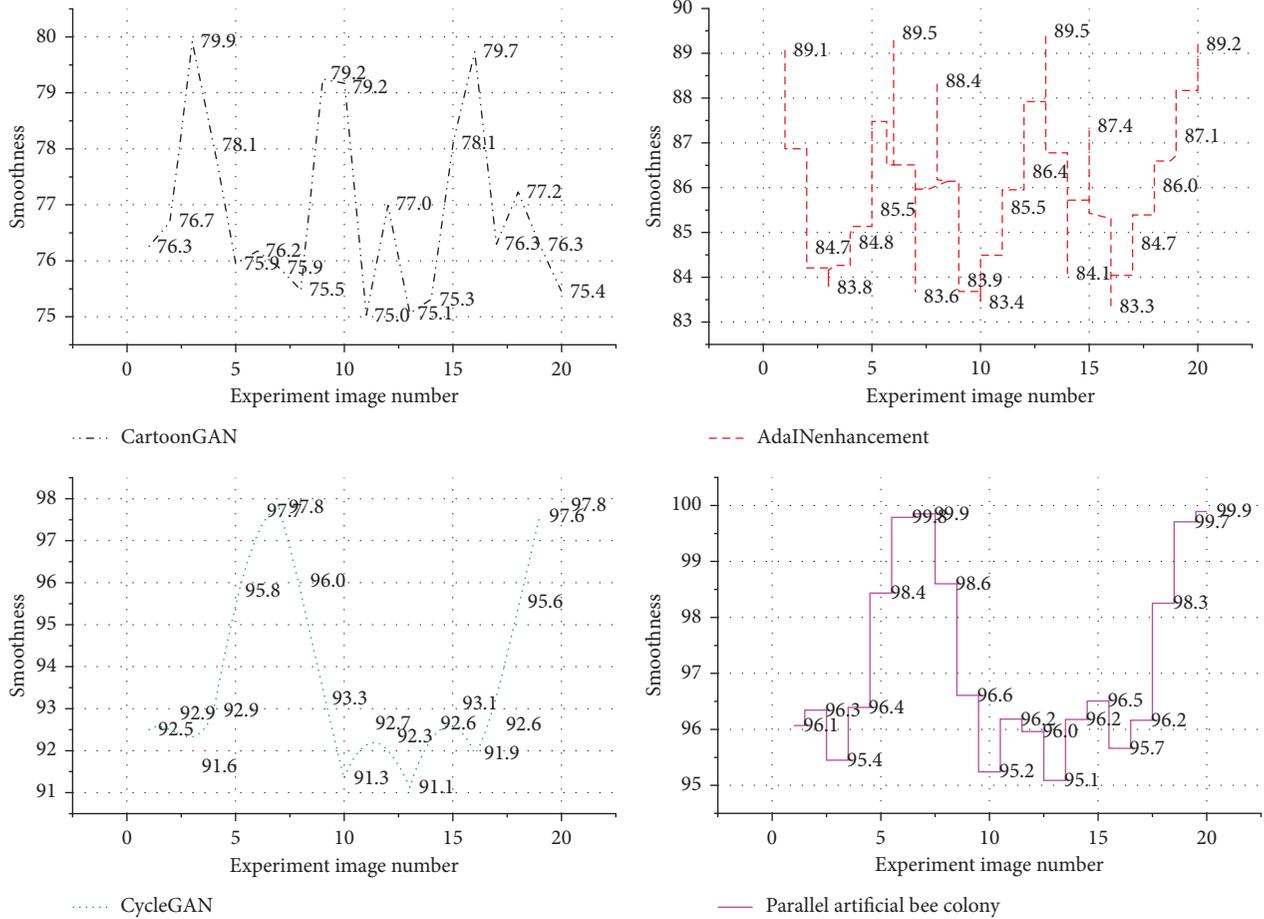


FIGURE 6: A sample of the output result of the HD rendering network.

TABLE 2: Comparison table of model size and parameter quantity of each method.

Method	Parallel artificial bee colony	CartoonGAN	AdaIN	CycleGAN
Parameter	6.91	6.33	6.81	6.42
Model size	31.10	24.03	27.02	25.11

enough. The output result pipeline image is very different from the original input pipeline image. In the experiment of this article, all cartoon pipeline images in the pipeline image training set are regarded as coming from the same generalized style, and the CartoonGAN model is trained based on this.

In order to quantitatively evaluate the method in this paper, we score the experimental results. For the pipeline image quality scoring experiment, we prepared 30 sets of test results, each of which is the output results of the four models

under the same input. For the style degree scoring experiment, we also prepared 30 sets of data pairs of pipeline images and stylized result pipeline images. Figure 7 is a comparison of the scoring results of each method. The specific value of the score is obtained by averaging all samples of each method and accurate to one decimal place. It can be seen that the parallel artificial bee colony algorithm achieved the highest score in the pipeline image quality score. In addition, we compared the running time of each method, as shown in Figure 8.

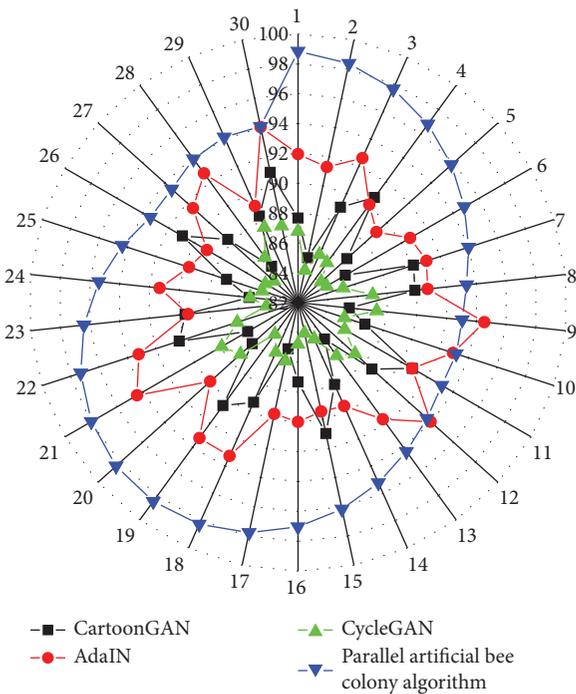


FIGURE 7: Comparison of scoring results of various methods.

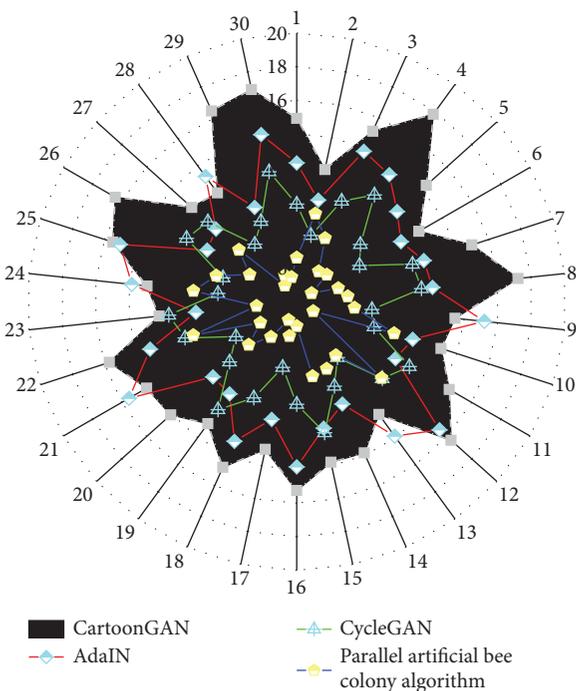


FIGURE 8: Running time of each method.

5. Conclusion

This paper studies the principles and methods of parallel algorithm design based on the CPU-GPU heterogeneous computing architecture. By analyzing the inherent parallelism of swarm intelligence algorithms, a general parallel processing framework for swarm intelligence optimization algorithms is proposed. The three-dimensional pipeline automatic

modeling method based on the ArcGIS platform is researched, and the establishment method of automatically building a three-dimensional pipeline model using the data obtained from the detection is realized. Through bending pipe processing and texture mapping, a good three-dimensional model is obtained. Based on the understanding of the characteristics of the pipeline image itself, combined with the analysis of the shortcomings of the existing methods, this paper proposes a new deep learning-based pipeline image high-definition rendering solution, which can render any input pipeline image into a styled pipeline image. The solution is divided into three processing stages, which are the pre-processing of the pipeline image, the style conversion of the pipeline image after the processing, and the enhancement of the effect of the style conversion result image. In this paper, by disassembling the difficult high-definition rendering task of pipeline image into three subprocesses, the processing difficulty of each subtask is reduced to a limited extent, and higher-quality processing results are obtained. In the pre-processing part of the pipeline image, we performed the pipeline image enhancement operation and the edge-preserving filtering operation on the input pipeline image, which prompted the pixel-level distribution method of the pipeline image to be changed, making it suitable for subsequent style conversion tasks. The task of converting it into a style pipeline image is difficult, and the effect of high-definition rendering is improved at the same time. In the part of pipeline image style conversion, this paper proposes a new deep learning-based pipeline image high-definition rendering network, which is composed of three subnetworks: pipeline image feature modeling module, feature model alignment module, and pipeline image re-rendering module. We adjust the “feature model” of the input pipeline image according to the “feature model” of the reference style pipeline image to align it on the multiscale style features and the “feature model” obtained after the adjustment has semantic content similar to that of the input pipeline image. At the same time, it also has multiscale style features consistent with the style pipeline image used as a reference. Finally, the output pipeline image is generated from the obtained “feature model,” that is, the “feature model” is re-rendered back to the pipeline image space, and the cartoon rendering result of the pipeline image is obtained.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Youth Innovation project of Guangdong Province, Application research of Augmented Reality technology in Lingnan cultural creative packaging

design under the strategy of “Digital China,” No. 2021WQNCX153

References

- [1] Z. He, L. You, R. W. Liu, F. Yang, J. Ma, and N. Xiong, “A cloud-based real time polluted gas spread simulation approach on virtual reality networking,” *IEEE Access*, vol. 7, pp. 22532–22540, 2019.
- [2] S. Rasmuson, E. Sintorn, and U. Assarsson, “A low-cost, practical acquisition and rendering pipeline for real-time free-viewpoint video communication,” *The Visual Computer*, vol. 37, no. 3, pp. 553–565, 2021.
- [3] L. Shi, B. Li, C. Kim, P. Kellnhofer, and W. Matusik, “Towards real-time photorealistic 3D holography with deep neural networks,” *Nature*, vol. 591, no. 7849, pp. 234–239, 2021.
- [4] C. J. Ogayar-Anguita, A. J. Rueda-Ruiz, R. J. Segura-Sanchez, M. Diaz-Medina, and A. L. Garcia-Fernandez, “A GPU-based framework for generating implicit datasets of voxelized polygonal models for the training of 3D convolutional neural networks,” *IEEE Access*, vol. 8, pp. 12675–12687, 2020.
- [5] E. Karuzaki, N. Partarakis, N. Patsiouras et al., “Realistic virtual humans for cultural heritage applications,” *Heritage*, vol. 4, no. 4, pp. 4148–4171, 2021.
- [6] P. Fraga-Lamas, T. M. Fernandez-Carames, O. Blanco-Novoa, and M. A. Vilar-Montesinos, “A review on industrial augmented reality systems for the industry 4.0 shipyard,” *Ieee Access*, vol. 6, pp. 13358–13375, 2018.
- [7] C. Wu, B. Yang, W. Zhu, and Y. Zhang, “Toward high mobile GPU performance through collaborative workload off-loading,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 2, pp. 435–449, 2017.
- [8] J. Wang, Z. Sun, H. Ji, X. Zhang, T. Wang, and Y. Shen, “A fast 3D brain extraction and visualization framework using active contour and modern OpenGL pipelines,” *IEEE Access*, vol. 7, pp. 156097–156109, 2019.
- [9] J. Liu, P. L. Rosin, X. Sun, J. Xiao, and Z. Lian, “Image-driven unsupervised 3D model co-segmentation,” *The Visual Computer*, vol. 35, no. 6, pp. 909–920, 2019.
- [10] A. Fatemah, S. Rasool, and U. Habib, “Interactive 3D visualization of chemical structure diagrams embedded in text to aid spatial learning process of students,” *Journal of Chemical Education*, vol. 97, no. 4, pp. 992–1000, 2020.
- [11] S. N. Silva Junior, F. C. Chamone, R. C. Ferreira, and E. R. Nascimento, “A 3D modeling methodology based on a concavity-aware geometric test to create 3D textured coarse models from concept art and orthographic projections,” *Computers & Graphics*, vol. 76, pp. 73–83, 2018.
- [12] Z. Zhang, J. Li, Y. Guo, C. Yang, and C. Wang, “3D highway curve reconstruction from mobile laser scanning point clouds,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 11, pp. 4762–4772, 2019.
- [13] Q. Zhang, “Web-based medical data visualization and information sharing towards application in distributed diagnosis,” *Informatics in Medicine Unlocked*, vol. 14, pp. 69–81, 2019.
- [14] W. Song, S. Zou, Y. Tian, and S. Sun, “A CPU-GPU hybrid system of environment perception and 3D terrain reconstruction for unmanned ground vehicle,” *Journal of Information Processing Systems*, vol. 14, no. 6, pp. 1445–1456, 2018.
- [15] B. Zhou, V. Lau, and X. Wang, “Machine-learning-based leakage-event identification for smart water supply systems,” *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 2277–2292, 2019.
- [16] G. Galanakis, X. Zabulis, T. Evdaimon et al., “A study of 3D digitisation modalities for crime scene investigation,” *Forensic Science*, vol. 1, no. 2, pp. 56–85, 2021.
- [17] N. Senin, S. Catalucci, M. Moretti, and R. K. Leach, “Statistical point cloud model to investigate measurement uncertainty in coordinate metrology,” *Precision Engineering*, vol. 70, pp. 44–62, 2021.
- [18] G. Alomía, D. Loaiza, C. Zúñiga, and X. Luo, “Procedural modeling applied to the 3D city model of bogota: a case study,” *Virtual Reality & Intelligent Hardware*, vol. 3, no. 5, pp. 423–433, 2021.
- [19] A. P. Cláudio, M. B. Carmo, A. A. de Carvalho, and W. Xavier, “Recreating a medieval urban scene with virtual intelligent characters: steps to create the complete scenario,” *Virtual Archaeology Review*, vol. 8, no. 17, pp. 31–41, 2017.
- [20] S. Chen, Y. Chen, S. Zhang, and N. Zheng, “A novel integrated simulation and testing platform for self-driving cars with hardware in the loop,” *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 3, pp. 425–436, 2019.